

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 5月 1日

出 願 番 号

Application Number:

特願2000-132694

出 願 人

Applicant (s):

株式会社東芝

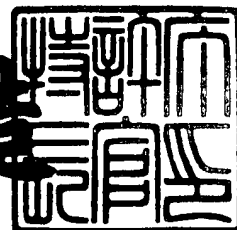


CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 3月 9日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3017968

【書類名】 特許願

【整理番号】 A000002470

【提出日】 平成12年 5月 1日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 3/00

【発明の名称】 画面制御プログラム及び動的表示情報取得プログラム並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体

【請求項の数】 6

【発明者】

 【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝府中工場内

 【氏名】 今村 大輔

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【代理人】

 【識別番号】 100058479

 【弁理士】

 【氏名又は名称】 鈴江 武彦

 【電話番号】 03-3502-3181

【選任した代理人】

 【識別番号】 100084618

 【弁理士】

 【氏名又は名称】 村松 貞男

【選任した代理人】

 【識別番号】 100068814

 【弁理士】

 【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画面制御プログラム及び動的表示情報取得プログラム並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体

【特許請求の範囲】

【請求項 1】 コンピュータに、

画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる動的表示情報取得機能と、

前記動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理に対して該当する前記表示属性情報を提供させる属性情報提供機能と

を実現させるための画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 2】 請求項 1 記載の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記属性情報提供機能は、前記動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理を組み込むことを特徴とする画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 3】 請求項 1 又は請求項 2 記載の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記少なくとも一つの動的表示処理のいずれかは、表示属性情報の内容に基づいて検索を行う処理であることを特徴とする画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 4】 コンピュータに、

画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識

別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる取得機能と、

前記動的表示処理に対して該当する表示属性情報を提供する機能に対して、前記取得機能によって取得された動的表示処理識別情報と表示属性情報とを提供させる提供機能と

を実現させるための動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 5】 請求項 4 記載の動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記動的表示処理識別情報と前記表示属性情報とは、所定のフォーマットで記述されており、

コンピュータに、

前記取得機能と前記提供機能との間において、前記所定のフォーマットで記述されている動的表示処理識別情報と表示属性情報とを、前記動的表示処理に対して該当する表示属性情報を提供する機能で取り扱い可能なフォーマットに変換させるフォーマット変換機能を付加した動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 6】 コンピュータに、

表示画面に対する入力内容が正常な時点で、この正常な入力内容を画面表示トランザクション情報として記憶させる記憶機能と、

再表示指示を受け付けた場合に、前記記憶機能によって記憶された画面表示トランザクション情報を参照させ、任意の時点での正常な入力内容を前記表示画面に対して再現させる再現機能と

を実現させるための画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ディスプレイに表示され、ユーザが対話的に入力する画面を提供す

るための画面制御プログラム、及び動的表示情報取得プログラム、並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】

ユーザが対話的にデータを入力するための画面プログラムを作成する場合において、従来はプログラム言語で画面プログラムを記述して作成する方法が利用されている。

【0003】

また、他の画面プログラムの作成方法として、開発ツールを用いて画面イメージを作画的に編集し画面に関するプログラムソースを生成する方法や、ファイルなどに画面のレイアウト情報を別途定義して実行時にプログラムがこの画面のレイアウト情報をロードして表示する方法も利用されている。

【0004】

【発明が解決しようとする課題】

上記の3種類の画面プログラム作成方法のうち、プログラム言語により画面プログラムを記述する方法を利用すると、プログラムの自由度は高くなるが記述量が多くなる。また、プログラムを変更する場合にプログラムの編集及びコンパイル等の作業が必要であり、開発効率の向上が困難である。

【0005】

また、開発ツールを用いて画面プログラムを作成する方法を利用した場合、画面のレイアウト作成は容易であるが、画面表示ために必要な処理に関する部分をプログラム言語で記述する必要がある、画面プログラムの作成が困難な場合がある。また、この方法ではレイアウトの変更が必要な場合にコンパイルする必要がある点も上記の場合と同様である。

【0006】

画面のレイアウト情報を定義して実行時にロードする方法を利用した場合には、画面のレイアウトを変更する場合であってもプログラムソースの編集とコンパイルが不要であるためプログラムの開発効率は高い。しかしながら、この方法で

は、プログラム外部に定義できるレイアウト情報の情報量に限りがあり、自由度と拡張性が高くない。例えば、画面を構成する部品の大きさ、位置、色の種類、画面上に配置される文字の指定などに関してはレイアウト情報として定義可能であるが、画面表示のために必要な処理に関する部分はプログラム内部で定義する必要がありプログラム外部で定義されない。

【 0 0 0 7 】

すなわち、上記 3 種類の画面プログラム作成方法を利用する場合には、画面表示のために必要な処理に関する部分をプログラム言語によりプログラミングする必要があり、また変更時にコンパイルする必要があるため、開発労力がかかるという問題がある。

【 0 0 0 8 】

加えて、業務アプリケーションで問題になるのは、障害発生時のアプリケーションの状態管理である。業務処理の障害に応じてデータベースの内容を復帰させることは通常行われているが、画面についても同様に復帰が望まれる場合がある。業務アプリケーションが障害から復帰した場合に、障害発生前において入力済み内容を、ユーザが画面に対して再入力するのは効率的ではない。

【 0 0 0 9 】

しかしながら、画面の表示状態を適切な状態に復帰させるプログラム開発には多くの労力がかかる。

【 0 0 1 0 】

本発明は、以上のような実情に鑑みてなされたもので、拡張性の高い画面プログラムを効率的かつ容易に開発し、画面プログラムの耐障害性を向上させるための画面制御プログラム、及び動的表示情報取得プログラム、並びに画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【 0 0 1 1 】

【課題を解決するための手段】

本発明の骨子は、画面に表示される場合に動的に変化する表示内容を得るために少なくとも一つの動的表示処理が予め作成されており、この動的表示処理の指

定及びこの動的表示処理に提供する属性の指定によって画面を構成可能な点にある。

【 0 0 1 2 】

本発明を実現するにあたって講じた具体的手段について以下に説明する。

【 0 0 1 3 】

第 1 の発明は、コンピュータに、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる動的表示情報取得機能と、動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理に対して該当する表示属性情報を提供させる属性情報提供機能とを実現させるための画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【 0 0 1 4 】

この第 1 の発明における画面制御プログラムを利用すると、ユーザは、表示する時点で動的に変化する内容についても、画面プログラム開発時あるいは変更時に動的表示処理の指定と表示属性情報の指定を行うのみで画面上に表示させることができる。

【 0 0 1 5 】

したがって、画面表示のために必要な処理に関する部分をプログラム言語によりプログラミングする必要がなく、容易に画面プログラムを作成でき、開発時あるいは変更時にコンパイルする必要もない。

【 0 0 1 6 】

これにより、ユーザは拡張性の高い画面プログラムを効率的に開発でき、画面プログラムを容易に変更できる。

【 0 0 1 7 】

第 2 の発明は、第 1 の発明と同様の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であるが、属性情報提供機能は、動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理を組

み込むという特徴を持つ。

【 0 0 1 8 】

すなわち、この第2の発明においては、予め作成済みの動的表示処理にうち、指定されたものが属性情報提供機能に対してプラグインされ、表示属性情報に基づく利用が可能となる。

【 0 0 1 9 】

これにより、ユーザは画面プログラム開発時あるいは変更時にプログラミングする必要がない。すなわち、ユーザは、単に動的表示処理と表示属性情報を指定するのみで画面プログラムにおける処理に関する部分についても開発あるいは変更可能となる。

【 0 0 2 0 】

第3の発明は、第1又は第2の発明と同様の画面制御プログラムを記録したコンピュータ読み取り可能な記録媒体であり、少なくとも一つの動的表示処理のいずれかは、表示属性情報の内容に基づいて検索を行う処理であるという特徴を持つ。

【 0 0 2 1 】

これにより、あるキーにしたがって検索されたデータを画面に表示させる場合であっても、ユーザはこの検索を行う動的表示処理を指定し、そのキーを表示属性情報として指定するのみで、検索結果を画面上に表示できる。したがって、上記と同様の効果を得ることができる。

【 0 0 2 2 】

なお、動的表示処理の他の例としては、入力内容をチェックしてエラーの場合にその旨を返す入力チェック処理、新規の画面をポップアップするポップアップ処理などがある。

【 0 0 2 3 】

第4の発明は、コンピュータに、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理のうちのいずれかを指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる取得機能と、動

的表示処理に対して該当する表示属性情報を提供する機能に対して、取得機能によって取得された動的表示処理識別情報と表示属性情報とを提供させる提供機能とを実現させるための動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【 0 0 2 4 】

この第4の発明における動的表示情報取得プログラムは、先の第1の発明における画面制御プログラムの動的表示情報取得機能に対応するプログラムである。

【 0 0 2 5 】

第5の発明は、第4の発明と同様の動的表示情報取得プログラムを記録したコンピュータ読み取り可能な記録媒体であるが、動的表示処理識別情報と表示属性情報とは所定のフォーマットで記述されており、コンピュータに、取得機能と提供機能との間において、所定のフォーマットで記述されている動的表示処理識別情報と表示属性情報とを、動的表示処理に対して該当する表示属性情報を提供する機能で取り扱い可能なフォーマットに変換させるフォーマット変換機能を付加した特徴を持つ。

【 0 0 2 6 】

これにより、例えばXML (Extensible Markup Language) ファイル、HTML (HyperText Markup Language) ファイル、CSVファイル等のような所定の形式で記述されたファイルとして動的表示処理識別情報と表示属性情報とを受け付け、その内容を後段の機能で取り扱い可能に変換し、提供できる。

【 0 0 2 7 】

第6の発明は、コンピュータに、表示画面に対する入力内容が正常な時点で、この正常な入力内容を画面表示トランザクション情報として記憶させる記憶機能と、再表示指示を受け付けた場合に、記憶機能によって記憶された画面表示トランザクション情報を参照させ、任意の時点での正常な入力内容を表示画面に対して再現させる再現機能とを実現させるための画面表示トランザクションプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【 0 0 2 8 】

この第6の発明における画面表示トランザクションプログラムは、上記の動的

表示処理の一つとして動作可能である。この画面表示トランザクションプログラムを利用すると、画面への入力作業中に障害が発生しても、障害から復帰した場合に障害発生前の任意の時点での正常な入力内容が画面上に再表示される。

【 0 0 2 9 】

したがって、ユーザの操作が効率化され、画面プログラムの耐障害性を容易に向上させることができる。

【 0 0 3 0 】

上記のようなプログラムを記録した記録媒体を用いることによって、上述した機能を有していない計算機システム、サーバやクライアント等の計算機に対して簡単に上述した機能を付加することができる。

【 0 0 3 1 】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施の形態について説明する。

【 0 0 3 2 】

（第 1 の実施の形態）

本実施の形態においては、レイアウト情報に加えて、画面表示のために必要な処理に関する部分を指定する動的表示情報を取得し、アプリケーション・プログラム（以下、「アプリケーション」という）の画面を表示する画面制御プログラムについて説明する。

【 0 0 3 3 】

図 1 は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図である。

【 0 0 3 4 】

本実施の形態に係る画面制御プログラム 1 は、主にレイアウト情報取得機能 2、動的表示情報取得機能 3、制御機能 4、検索処理インタフェース 5 a、入力チェック処理インタフェース 5 b、ポップアップ処理インタフェース 5 c、データ入出力機能 6、画面表示トランザクション機能 7 を計算機や計算機システム上で提供する。

【 0 0 3 5 】

また、この画面制御プログラム 1 は、例えば検索処理や、入力チェック処理、ポップアップ処理などのように、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている動的表示処理（以下、「コンポーネント」という）8 a ～ 8 c を利用する。コンポーネント 8 a ～ 8 c は既にコンパイルされており、そのままの状態で行実行可能なプログラムである。

【 0 0 3 6 】

ユーザ（画面表示プログラム作成者）は、アプリケーション 1 1 の画面についてのレイアウト情報 9 と動的表示情報 1 0 とを入力する。

【 0 0 3 7 】

レイアウト情報 9 には、先に述べたように、例えば画面を構成するための部品の配置位置、大きさ、色などのような、画面を構成する場合に固定的に決定される静的情報が主に記述されている。

【 0 0 3 8 】

動的表示情報 1 0 には、ユーザが利用したいコンポーネントを指定した動的表示処理識別情報や、このコンポーネントの処理内容を決定するために必要な表示属性情報が記述されている。ここでは、例として検索処理、入力チェック処理、ポップアップ処理を指定する旨の動的表示処理識別情報が記述されており、また検索処理において図示しないテーブル T_1 の内容が検索対象となる旨の表示属性情報が記述されている。

【 0 0 3 9 】

レイアウト情報取得機能 2 は、レイアウト情報 9 に定義された内容を解析し、解析結果を制御機能 4 に伝える。

【 0 0 4 0 】

動的表示情報取得機能 3 は、動的表示情報 1 0 に定義された内容を解析し、解析結果を制御機能 4 に伝える。この動的表示情報取得機能 3 は、主に取得機能 3 a と、変換機能 3 b と、提供機能 3 c とから構成されている。

【 0 0 4 1 】

所定のフォーマットで記述されている動的表示情報 1 0 が取得機能 3 a によって取り込まれると、その記述内容は変換機能 3 b によって制御機能 4 で取り扱い

可能なフォーマットに変換され、提供機能 3 c によって制御機能 4 に提供される。

【 0 0 4 2 】

制御機能 4 の備える属性情報提供機能 4 a は、動的表示情報取得機能 3 から提供された動的表示処理識別情報によって指定されているコンポーネントをプラグインするための制御を行い、プラグインされたコンポーネントに対して該当する表示属性情報を提供し、結果を取得する。

【 0 0 4 3 】

また、制御機能 4 は、画面上への入出力用のフィールド（画面上で入出力を行うための部品）の表示や動作の制御を行う。

【 0 0 4 4 】

検索処理インタフェース 5 a は、検索処理用のコンポーネント 8 a をプラグインし、動的表示情報 1 0 に定義された検索用の表示属性情報に基づいてファイルやデータベースのテーブルを検索し、検索結果を制御機能 4 に返却する。

【 0 0 4 5 】

入力チェック処理インタフェース 5 b は、入力チェック処理用のコンポーネント 8 b をプラグインし、レイアウト情報 9 もしくは動的表示情報 1 0 に定義された入力チェック用の表示属性情報に基づいてフィールドに入力された値をチェックし、その結果を制御機能 4 に返却する。

【 0 0 4 6 】

ポップアップ処理インタフェース 5 c は、ポップアップ処理用のコンポーネント 8 c をプラグインし、動的表示情報 1 0 に定義されたポップアップ処理用の表示属性情報に基づいてポップアップウィンドウを表示し、ポップアップウィンドウにて決定されたデータを制御機能 4 に反映する。

【 0 0 4 7 】

データ入出力機能 6 は、画面上のフィールドにおいて入力されたデータを取得し、またアプリケーション 1 1 から提供されたデータを画面上のフィールドにセットする。

【 0 0 4 8 】

画面表示トランザクション処理機能 7 は、入出力の状態が正常な場合にその入出力状態を示すオブジェクトを保存する。また、画面に対するトランザクションの開始を指示された場合に、保存されているオブジェクトを用いてトランザクションの失敗による画面表示の復帰を行う。

【 0 0 4 9 】

以下に、上記構成により実現される動作を具体的に説明する。

【 0 0 5 0 】

図 2 は、レイアウト情報 9 の具体例を示す図である。このレイアウト情報 9 の例では、XML を用いて記述されている場合を示しているが、その他にも CSV 形式、特定の文字をデリミタとする形式、固定長ファイル形式が利用できる。

【 0 0 5 1 】

図 3 は、本実施の形態に係る画面制御プログラム 1 がレイアウト情報 9 を読み込んだ場合の処理手順を例示する図である。

【 0 0 5 2 】

まず、アプリケーション 1 1 が、レイアウト情報取得機能 2 に対してレイアウト情報 9 をセットする (3 0 1) 。レイアウト情報取得機能 2 は、先の図 2 に示すレイアウト情報 9 の 2 行目 ~ 2 1 行目に示す <field> ~ </field> を 1 フィールド分の定義として展開する (3 0 2) 。XML 文書の展開には XML パーザといった既存の技術を用いることができる。

【 0 0 5 3 】

展開されたフィールド毎のレイアウト情報は、レイアウト情報取得機能 2 から制御機能 4 に対してセットされる (3 0 3) 。

【 0 0 5 4 】

制御機能 4 は、セットされたレイアウト情報に従ってフィールドを作成する (3 0 4) 。以下、処理 3 0 2 ~ 3 0 4 が <field> タグの数だけ繰り返される (3 0 5) 。

【 0 0 5 5 】

図 4 は、<field> タグの数だけ繰り返される各フィールドの作成処理を例示するフローチャートである。

【 0 0 5 6 】

制御機能 4 は、先の図 2 に示すレイアウト情報 9 の 2 行目の <field> タグ内に指定されている I D に関連付けたフィールドオブジェクトを作成する（4 0 1）。

【 0 0 5 7 】

次に、先の図 2 に示すレイアウト情報 9 の 3 行目～1 1 行目に定義されている位置属性などのフィールドの表示に関する情報を作成されたフィールドオブジェクトに対してセットする（4 0 2）。

【 0 0 5 8 】

例えば、制御機能 4 は、先の図 2 に示すレイアウト情報 9 の 7 行目の <label> タグに囲まれた属性により、フィールドのラベルをフィールドオブジェクトにセットする。

【 0 0 5 9 】

次に、制御機能 4 は、レイアウト情報 9 に入力チェック属性が定義されているか否かを判断する（4 0 3）。先の図 2 に示すレイアウト情報 9 については、1 3 行目の <validation> タグの有無により判断できる。

【 0 0 6 0 】

ここで、入力チェック属性が指定されている場合には、先の図 2 に示すレイアウト情報 9 の 1 5 行目の <component> タグで指定された入力チェック用のコンポーネントを作成するか、あるいはコンポーネントのインスタンスを取得する（4 0 4）。J a v a 言語の場合、クラス名からオブジェクトの作成が可能であり、この処理 4 0 4 に用いることができる。

【 0 0 6 1 】

次に、制御機能 4 は、入力チェック用コンポーネントをフィールドに登録する（4 0 5）。

【 0 0 6 2 】

次に、制御機能 4 は、先の図 2 に示すレイアウト情報 9 の 1 4 行目に示す入力チェックのトリガーとなるキーと、1 6 行目～1 9 行目に示すエラー時の動作とをフィールドに登録する（4 0 6）。

【 0 0 6 3 】

そして、制御機能 4 は、フィールドを先の図 2 に示すレイアウト情報 9 の 1 2 行目に指定するように初期状態にして完了する (4 0 7) 。この例では、初期状態は「UNEDITABLE」となっている。

【 0 0 6 4 】

図 5 は、上記のようなレイアウト情報 9 に基づく制御機能 4 の処理によって構成される画面を示す図である。

【 0 0 6 5 】

レイアウト情報 9 に基づいて作成された画面 1 2 は、ラベル 1 2 a ～ 1 2 c や初期状態のフィールド 1 2 d ～ 1 2 f などのような不変の画面要素を表示している。

【 0 0 6 6 】

例えば、先の図 2 に示すレイアウト情報 9 の 7 行目の <label> タグに囲まれた属性はフィールド 1 2 d のラベル 1 2 a を示しているが、表示されると「数量」となる。

【 0 0 6 7 】

また、先の図 2 に示すレイアウト情報 9 の 1 2 行目において、初期状態は「UNEDITABLE」とする旨が指定されているため、フィールド 1 2 d の初期状態は編集不可能状態となる。

【 0 0 6 8 】

図 6 は、画面制御プログラム 1 がレイアウト情報 9 を読み込んだ場合の処理手順の変形例を示す図である。

【 0 0 6 9 】

この変形例では、アプリケーション 1 1 が、レイアウト情報取得機能 2 に対してレイアウト情報 9 をセットし (6 0 1) 、 <field> ～ </field> まだが 1 フィールド分の定義として展開する (6 0 2) 。

【 0 0 7 0 】

次に、レイアウト情報取得機能 2 が 1 フィールド分のレイアウト情報の格納されたオブジェクトを作成し (6 0 3 、 6 0 4) 、そのオブジェクトを制御機能 4

にセットし（605）、フィールドが作成されて属性がセットされる（606）。

【0071】

その後、処理602から606までの動作がフィールドの数だけ繰り返される（607）。

【0072】

図7は、レイアウト情報9において定義された入力チェック属性に従って画面制御プログラム1が実行する入力チェック処理の処理手順を例示する図である。

【0073】

ユーザは、フィールドに値を入力した後、入力チェックのトリガーとなる操作を行う（701）。先の図2に示すレイアウト情報9の14行目に〈trigger〉タグで囲まれた属性として「VK_ENTER」が指定されているが、これはエンターキーの入力が入力チェックのトリガーになることを示している。

【0074】

このトリガーキーの入力によって、制御機能4は、フィールドに入力された値を入力チェック処理インタフェース5bに対して渡し、入力のチェックを指示する（702）。

【0075】

入力チェック処理インタフェース5bは、値が入力されたフィールドに登録された入力チェック処理用のコンポーネント8bに対して入力チェックを指示する（703）。

【0076】

入力チェック処理用のコンポーネント8bは、渡された値を自分の実装により判断し、エラーか否かを返却する（704）。

【0077】

入力チェック処理インタフェース5bは、得られた結果を制御機能4に返却する（705）。

【0078】

制御機能4は、チェック結果がエラーだった場合に先の図2に示す16行目～

1 9 行目に定義された内容に従ってフィールドの属性変更を行う。

【 0 0 7 9 】

このような動作は、フィールドを構成するプログラムに予め用意されており、レイアウト情報 9 が記述されているファイルでは、それらの中から選択的に指定している。この例では、`<type>` タグに「CHANGE_BACKGROUND_COLOR」と定義されており、また `<param>` タグで「background」が #FF0000 と指定されている。

【 0 0 8 0 】

したがって、図 8 に例示するように、入力チェックの結果がエラーだった場合には、そのフィールド 1 2 d の背景色が赤になる。

【 0 0 8 1 】

続いて、本実施の形態において特徴的な情報である動的表示情報 1 0 について説明する。なお、上記のレイアウト情報 9 において説明した入力チェック属性は、この動的表示情報 1 0 で指定されてもよい。

【 0 0 8 2 】

図 9 は、動的表示情報 1 0 の具体例を示す図である。

【 0 0 8 3 】

この動的表示情報 1 0 の例では、レイアウト情報 9 と同様に XML を用いて記述された場合を示しているが、他にも CSV 形式、特定の文字をデリミタとする形式、固定長ファイル形式で記述してもよい。

【 0 0 8 4 】

図 1 0 は、本実施の形態に係る画面制御プログラム 1 が動的表示情報 1 0 を読み込んだ場合の処理手順を例示する図である。

【 0 0 8 5 】

まず、アプリケーション 1 1 が、動的表示情報取得機能 3 に対して動的表示情報 1 0 をセットする (1 0 0 1)。動的表示情報取得機能 3 は、先の図 9 に示す動的表示情報 1 0 の 2 行目～1 3 行目に示す `<field>` ～ `</field>` を 1 フィールド分の定義として展開する (1 0 0 2)。XML 文書の展開にはレイアウト情報 9 の場合と同様に XML パーザといった既存の技術を用いることができる。

【 0 0 8 6 】

展開されたフィールド毎の動的表示情報は、動作表示情報取得機能 3 から制御機能 4 に対してセットされる（1 0 0 3）。

【0 0 8 7】

制御機能 4 は、セットされた動的表示情報に従ってフィールドの表示属性をセットする（1 0 0 4）。以下、処理 1 0 0 2～1 0 0 4 の処理が〈field〉タグの数だけ繰り返される（1 0 0 5）。この結果、画面上の個々のフィールドについての表示属性が変更され、かつ動作に必要な情報が各フィールドにセットされる。

【0 0 8 8】

図 1 1 は、上記のような動的表示情報 1 0 に基づく制御機能 4 の処理によって構成される画面を示す図である。

【0 0 8 9】

この画面 1 2 においては、先の図 9 に示す動的表示情報 1 0 の 3 行目に指定するように、「area」という ID のフィールドは〈state〉タグで囲まれた「COMBO」という表示属性になる。この「COMBO」という表示属性は、一覧からの選択を可能としかつテキストフィールドに対して直接の文字入力を可能とする旨を示す動的表示処理識別情報であり、画面要素 1 4 a に示すような動的な表示形態をとる。

【0 0 9 0】

また、先の図 9 に示す動的表示情報 1 0 の 1 5 行目に指定する「product」という ID を持つフィールドは「POPUP」という表示属性である。この「POPUP」はテキストフィールドにポップアップウィンドウを表示する時に押下するためのボタンを付属する旨を示す動的表示処理識別情報であり、画面要素 1 4 b に示すような動的な表示形態をとる。このフィールドにセットされる表示属性毎の処理については後述する。

【0 0 9 1】

図 1 2 は、画面制御プログラム 1 が動的表示情報 1 0 を読み込んだ場合の処理手順の変形例を示す図である。

【0 0 9 2】

この変形例では、アプリケーション 1 1 が、動的表示情報取得機能 3 に対して動的表示情報 1 0 をセットし (1 2 0 1)、`<field> ~ </field>` までを 1 フィールド分の定義として展開する (1 2 0 2)。

【0 0 9 3】

次に、動的表示情報取得機能 3 が 1 フィールド分の動的表示情報の格納されたオブジェクトを作成し (1 2 0 3、1 2 0 4)、そのオブジェクトを制御機能 4 にセットし (1 2 0 5)、フィールドへの属性がセットされる (1 2 0 6)。

【0 0 9 4】

そして、処理 1 2 0 2 から 1 2 0 6 と同様の動作がフィールドの数だけ繰り返される (1 2 0 7)。

【0 0 9 5】

図 1 3 は、フィールドにセットされた表示属性が検索だった場合に画面制御プログラム 1 が実行する処理の手順を例示する図である。

【0 0 9 6】

また、図 1 4 は、フィールドにセットされた表示属性が検索だった場合の画面制御プログラム 1 の処理を例示するフローチャートである。

【0 0 9 7】

制御機能 4 に動的表示情報 1 0 がセットされると、制御機能 4 は各フィールド毎にフィールドの表示属性を変更する (1 4 0 1)。

【0 0 9 8】

制御機能 4 は、フィールドの表示属性が「COMBO」、「PULLDOWN」といった一覧選択を行うものであるか否か (検索についての動的表示処理識別情報か否か) を判断し (1 4 0 2)、その場合は一覧に格納するデータを検索するのに必要な属性 (表示属性情報) が動的表示情報にあるか否かを判断する (1 4 0 3)。

【0 0 9 9】

検索に必要な属性がある場合には、検索処理インタフェース 5 a にテーブル名、検索するカラム名、検索条件といった検索のための属性を渡して検索の指示を出す (1 3 0 1)。

【0 1 0 0】

検索処理インタフェース 5 a は、プラグインされた検索処理用のコンポーネント 8 a を用いて検索処理を行う（1 3 0 2）。プラグインされた検索処理用のコンポーネント 8 a は、渡された検索のための属性を利用し自らの実装に基づいてデータベースやファイルのテーブルを検索する（1 3 0 3、1 4 0 4）。

【0 1 0 1】

検索結果は、検索処理用のコンポーネント 8 a から検索処理インタフェース 5 a を経由し（1 3 0 4）、制御機能 4 に渡され（1 3 0 5）、フィールドの一覧データに格納される（1 3 0 6）。

【0 1 0 2】

上記の処理の結果、図 1 5 に示すように、検索処理用のコンポーネント 8 a によって動的な表示内容が取得され、画面 1 2 のフィールド 1 2 e にユーザの操作に基づいて検索された一覧 1 5 a が表示可能となる。

【0 1 0 3】

図 1 6 は、フィールドにセットされた表示属性がポップアップだった場合に画面制御プログラム 1 が実行する処理の手順を例示する図である。

【0 1 0 4】

制御機能 4 に動的表示情報がセットされると、制御機能 4 は、検索処理の場合と同様にフィールドの表示属性を変更し、フィールドの表示属性がポップアップを示すものか否か（ポップアップについての動的表示処理識別情報か否か）を判断する。例えば、先の図 9 に示す動的表示情報 1 0 の 1 5 行目に指定する「POPUP」がポップアップを示す表示属性である。

【0 1 0 5】

表示属性がポップアップの場合には、ポップアップ処理インタフェース 5 c に対してポップアップウィンドウに表示するコンポーネント名、ポップアップウィンドウのタイトルバー部分に表示するタイトル、大きさ、ポップアップウィンドウに付属するボタンのラベル名といった属性（表示属性情報）を渡して、ポップアップ処理 8 c の登録を指示する（1 6 0 1）。

【0 1 0 6】

ポップアップ処理インタフェース 5 c に渡す属性は、先の図 9 に示す動的 1 8

行目～22行目に指定するように定義可能である。

【0107】

ポップアップ処理インタフェース5cは、プラグインされたポップアップ用のコンポーネント8cを作成、もしくはインスタンスを取得し（1602、1603）、取得したポップアップ用のコンポーネント8cの表示処理をフィールドへの操作の結果起動される処理として登録する（1604）。これはJ a v a 言語を例にとると、ボタン押下のイベントにより起動するモジュールをボタンに登録するという処理で実現可能である。

【0108】

図17は、ユーザによるフィールドへの操作の結果、起動されるポップアップ処理の手順を例示する図である。

【0109】

ユーザが制御機能4により表示されているフィールドに対しポップアップ表示を命ずる操作を行う（1701）。

【0110】

制御機能4は、ポップアップ処理インタフェース5cに対し、ポップアップウィンドウの表示を指示する（1702）。

【0111】

ポップアップ処理インタフェース5cは、ポップアップウィンドウを表示しウィンドウ上にプラグインされたポップアップ処理用のコンポーネント8cを表示する（1703）。

【0112】

この状態での画面の具体例を図18に示す。この図18における画面12の要素16aがプラグインされたポップアップ処理用のコンポーネント8cである。

【0113】

ここで、ユーザからの入力操作を待ち、ユーザはポップアップウィンドウに対してデータを入力する（1704）。図18の例では、画面要素16bにおける選択操作がこの入力操作に該当する。

【0114】

ユーザによってポップアップウィンドウに付属している決定ボタンの押下といった確定操作が行われると（１７０５）、ポップアップ処理インタフェース５ｃはポップアップウィンドウを閉じる（１７０６）。

【０１１５】

ポップアップウィンドウに付属している決定ボタンとは、図１８の画面要素１６ｃのように、ポップアップウィンドウ自体への操作のために予め用意されているボタンである。

【０１１６】

そして、ポップアップ処理インタフェース５ｃは、プラグインされたポップアップ処理用のコンポーネント８ｃに対しユーザが入力したデータを取得するための指示を発行する（１７０７）。

【０１１７】

その結果、ポップアップ処理用のコンポーネント８ｃは、入力されたデータのＩＤとデータの値をポップアップ処理インタフェース５ｃに返却する（１７０８）。ここでデータのＩＤとは、ポップアップ処理インタフェース５ｃとポップアップ処理用のコンポーネント８ｃとの間の規約により必要とされるもので、制御機能４の任意のフィールドにつけられたＩＤと同値である。

【０１１８】

ポップアップ処理インタフェース５ｃは、得られたデータを反映するフィールドをデータのＩＤをキーにして検索し、制御機能４の該当するフィールドにセットする（１７０９）。

【０１１９】

この状態での画面の具体例を図１９に示す。この図１９における画面１２のフィールド１２ｃには、ユーザがポップアップウィンドウで選択したデータがセットされ、表示される。

【０１２０】

図２０は、画面表示トランザクション機能７を利用して画面表示を復帰させる処理の手順を例示する図である。

【０１２１】

アプリケーション 1 1 は、画面表示トランザクション機能 7 に対して画面表示トランザクションの開始を指示する（2 0 0 1）。

【0 1 2 2】

画面表示トランザクション機能 7 は、制御機能 4 に対し、現在の画面状態の取得を指示する（2 0 0 2）。

【0 1 2 3】

制御機能 4 は、現在の画面状態を示す情報として、セットされているレイアウト情報、セットされている動的表示情報、各フィールドの活性データを格納した画面表示トランザクション情報オブジェクトを作成する（2 0 0 3、2 0 0 4）。ここで、活性データとは、ユーザがフィールドに入力したデータやアプリケーションの実行によってフィールドに表示されたデータを指す。

【0 1 2 4】

制御機能 4 は、画面表示トランザクション情報オブジェクトを画面表示トランザクション機能 7 に返却し（2 0 0 5）、画面表示トランザクション機能 7 はアプリケーション 1 1 に画面表示トランザクション情報オブジェクトを返却する（2 0 0 6）。

【0 1 2 5】

この時点での画面表示トランザクション情報オブジェクトは、表 1 に示すようにレイアウト情報、動的表示情報、活性データを管理している。

【0 1 2 6】

【表 1】

レイアウト情報	[レイアウトオブジェクト]
動的表示情報	[動的表示情報オブジェクト]
活性データ	[活性データの一覧オブジェクト]

【0 1 2 7】

なお、画面表示トランザクション情報オブジェクト自体の返却に代えて、画面

表示トランザクション情報オブジェクトを画面表示トランザクション機能 7、もしくは制御機能 4 によって特定の ID と対応づけて管理し、アプリケーション 11 に ID のみを返却するとしてもよい。

【0 1 2 8】

その後、アプリケーション 11 は、業務処理を続け（2 0 0 7）、業務処理が失敗し画面状態の復帰が必要になった場合に、画面表示トランザクション機能 7 に対しトランザクションの復帰を指示する（2 0 0 8）。この時、アプリケーション 11 は、画面表示トランザクション情報オブジェクト、もしくは画面表示トランザクション情報オブジェクトを示す ID を画面表示トランザクション機能 7 に渡して復帰を指示する。

【0 1 2 9】

画面表示トランザクション機能 7 は、渡された画面表示トランザクション情報オブジェクト、もしくは画面表示トランザクション情報オブジェクトを示す ID を制御機能 4 にセットして状態の復帰を指示する（2 0 0 9）。

【0 1 3 0】

制御機能 4 は、画面表示トランザクション情報オブジェクトがセットされると、画面表示トランザクション情報オブジェクトからレイアウト情報、動的表示情報、活性データに基づき、画面の状態をトランザクションが開始された時点の状態を含んだ任意の状態へ復帰する（2 0 1 0）。または、制御機能 4 は、画面表示トランザクション情報オブジェクトを示す ID がセットされると、ID から画面表示トランザクション情報オブジェクトを検索し、以下同様の手順で画面の状態を復帰する。

【0 1 3 1】

以上説明したように、本実施の形態に係る画面制御プログラム 1 を利用することにより、ユーザは、表示を要求する時点毎に動的に変化する表示内容についてもこの動的な表示内容を取得するために予め設けられている動的表示処理を指定し、またこの動的な表示内容を得るための属性を指定するのみで、画面上に表示させることができる。

【0 1 3 2】

したがって、表示内容が変化する画面プログラムを開発・変更する場合であってもプログラミングする必要がなく、コンパイルする必要もない。

【 0 1 3 3 】

ゆえに、画面プログラムを容易かつ効率的に開発することができ、開発労力を軽減できる。

【 0 1 3 4 】

また、本実施の形態に係る画面制御プログラム 1 を利用することにより、アプリケーション 1 1 が障害から復帰した場合に画面に障害発生前の正常な入力状態を再現させることができる。

【 0 1 3 5 】

したがって、ユーザは画面に再入力する必要がなく、入力操作を効率化させることができる。また、ユーザはこのような入力操作を効率化させる画面プログラムを容易に開発できる。

【 0 1 3 6 】

なお、本実施の形態においては、画面制御プログラム 1 が画面表示トランザクション機能 7 を備えている場合について説明しているが、これに限定されるものではなく、画面表示トランザクション機能を実現するためのコンポーネントをプラグインする形態を採用してもよい。

【 0 1 3 7 】

本実施の形態においてレイアウト情報取得機能 2 及び動的表示情報取得機能 3 は、画面を表示する際にそれぞれレイアウト情報 9 及び動的表示情報 1 0 を取得し、変換し、変換結果を制御機能 4 に提供する。しかしながら、これに代えて、レイアウト情報取得機能 2 及び動的表示情報取得機能 3 は、それぞれレイアウト情報 9 及び動的表示情報 1 0 を予め取得、変換しておき、画面を表示する際に制御機能 4 がこの変換済みの情報を読み込むとしてもよい。

【 0 1 3 8 】

（第 2 の実施の形態）

本実施の形態においては、レイアウト情報のセットを複数回に分けて行う画面制御プログラムについて説明する。

【 0 1 3 9 】

図 2 1 は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図であり、図 1 と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【 0 1 4 0 】

この画面制御プログラム 1 7 におけるレイアウト情報取得機能 1 8 は、レイアウト情報のセットを複数回に分けて行うことに対応している。このレイアウト情報取得機能 1 8 は、複数に分割されたレイアウト情報 1 9 を入力する。

【 0 1 4 1 】

図 2 2 は、本実施の形態に係る画面制御プログラム 1 7 がレイアウト情報を読み込んだ場合の処理手順を例示する図である。

【 0 1 4 2 】

処理 2 2 0 1 ～ 2 2 0 5 は、先の図 3 における処理 3 0 1 ～ 3 0 5 と同様である。これにより制御機能 4 に対してレイアウト情報が設定され、画面にはフィールドが初期状態で表示される。

【 0 1 4 3 】

次に、分割された別のレイアウト情報を指定してレイアウト情報の追加の指示がアプリケーション 1 1 よりレイアウト情報取得機能 1 8 に対して行われる（2 2 0 6）。

【 0 1 4 4 】

レイアウト情報取得機能 1 8 は、処理 2 2 0 2 と同様の手順でフィールドの属性情報を展開し（2 2 0 7）、制御機能 4 に対してレイアウト情報の追加を指示する（2 2 0 8）。

【 0 1 4 5 】

制御機能 4 は、追加されたフィールドのオブジェクトを作成し、属性をセットする（2 2 0 9）。以下、処理 2 2 0 7 ～ 2 2 0 8 が追加されたフィールドの数だけ繰り返される（2 2 1 0）。以上により必要に応じたレイアウトの初期化が可能になる

本実施の形態に係る画面制御プログラム 1 7 を利用しても、上記第 1 の実施の

形態と同様の作用効果を得ることができる。

【 0 1 4 6 】

(第 3 の実施の形態)

本実施の形態においては、レイアウト情報、動的表示情報がネットワーク上に配置された環境で利用される画面制御プログラムについて説明する。

【 0 1 4 7 】

図 2 3 は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図であり、図 1 と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【 0 1 4 8 】

画面制御プログラム 2 0 を備える計算機 2 1 は、ネットワーク 2 2 を介して W W W サーバ 2 3 と接続されている。

【 0 1 4 9 】

レイアウト情報取得機能 2 4 は、レイアウト情報 9 をネットワーク 2 2 からダウンロードすることに対応している。

【 0 1 5 0 】

動的表示情報取得機能 2 5 は、動的表示情報 1 0 をネットワーク 2 2 からダウンロードすることに対応している。

【 0 1 5 1 】

W W W サーバ 2 3 は、レイアウト情報 9、動的表示情報 1 0 のダウンロードを可能とするサーバプロセスの具体例である。

【 0 1 5 2 】

レイアウト情報 9 をネットワーク 2 2 からダウンロードする場合には、先の図 3 に示す処理 3 0 1 においてレイアウト情報 9 を U R L (Uniform Resource Locator) で指定する。

【 0 1 5 3 】

レイアウト情報取得機能 2 4 は、W W W サーバ 2 3 にアクセスして指定された U R L へのコネクションを確立し、レイアウト情報 9 を読み込む。

【 0 1 5 4 】

動的表示情報 10 をネットワーク 22 からダウンロードする場合もレイアウト情報 9 の場合と同様に行われ、先の図 10 に示す処理 1001 において動的表示情報 10 を URL で指定する。

【 0 1 5 5 】

J a v a 言語では URL で指定されたファイルを扱うクラスが用意されており、このクラスを用いることで実現可能である。

【 0 1 5 6 】

本実施の形態に係る画面制御プログラム 20 を利用しても、上記第 1 の実施の形態と同様の作用効果を得ることができる。

【 0 1 5 7 】

(第 4 の実施の形態)

本実施の形態においては、各インタフェースにプラグインされるコンポーネントをネットワーク上に分散させた環境で利用される画面制御プログラムについて説明する。

【 0 1 5 8 】

図 24 は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図であり、図 1 と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【 0 1 5 9 】

画面制御プログラム 26 は、図示しない計算機上で動作する。画面制御プログラム 26 における検索処理インタフェース 27 は、検索処理スタブ 28 と接続されている。

【 0 1 6 0 】

この検索処理スタブ 28 は、検索処理インタフェース 27 にプラグインされる検索処理用のコンポーネントであり、ネットワーク 29 を介して検索処理リモートオブジェクト 30 に検索処理を依頼し、検索処理リモートオブジェクト 30 が行った検索処理結果を受け取り、検索処理インタフェース 27 に返却する。

【 0 1 6 1 】

同様に、入力チェック処理インタフェース 31 は、入力チェック処理スタブ 3

2 と接続されている。

【0 1 6 2】

この入力チェック処理スタブ 3 2 は、入力チェック処理インタフェース 3 1 にプラグインされる入力チェック処理用のコンポーネントであり、ネットワーク 2 9 を介して入力チェック処理リモートオブジェクト 3 3 に入力チェック処理を依頼し、入力チェック処理リモートオブジェクト 3 3 が行った入力チェック処理結果を受け取り、入力チェック処理インタフェース 3 1 に返却する。

【0 1 6 3】

本実施の形態に係る画面制御プログラム 2 6 を利用しても、上記第 1 の実施の形態と同様の作用効果を得ることができる。

【0 1 6 4】

(第 5 の実施の形態)

本実施の形態においては、画面表示トランザクション情報がサーバに保持され、クライアントの状態をサーバから制御する画面制御プログラムについて説明する。

【0 1 6 5】

図 2 5 は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図であり、図 1 と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0 1 6 6】

画面制御プログラム 3 4 は、図示しないクライアント上で動作する。画面制御プログラム 3 4 における画面表示トランザクション機能 3 5 は、画面表示トランザクション情報送受信部 3 6 に接続されている。

【0 1 6 7】

画面表示トランザクション情報送受信部 3 6 は、画面表示トランザクション機能 3 5 から画面表示トランザクション情報（画面の状態情報）を取得し、ネットワーク 3 7 を介して画面表示トランザクション管理サーバ 3 8 に送信する。また、画面表示トランザクション管理サーバ 3 8 から受信した画面表示トランザクション情報を画面表示トランザクション機能 3 5 にセットする。

【 0 1 6 8 】

画面表示トランザクション管理サーバ 3 8 は、画面表示トランザクション送受信部 3 6 から受信した画面表示トランザクション情報をクライアント毎に管理し、必要に応じて取出し画面表示トランザクション情報送受信部 3 6 に送信する。

【 0 1 6 9 】

本実施の形態に係る画面制御プログラム 3 4 を利用しても、上記第 1 の実施の形態と同様の作用効果を得ることができる。

【 0 1 7 0 】

（第 6 の実施の形態）

本実施の形態においては、業務処理をプラグインする画面制御プログラムについて説明する。

【 0 1 7 1 】

図 2 6 は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図であり、図 1 と同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【 0 1 7 2 】

画面制御プログラム 3 9 の制御機能 4 0 は、業務処理インタフェース 4 1 と接続されている。

【 0 1 7 3 】

業務処理用のコンポーネント 4 2 は、他の画面表示に関わるコンポーネントと同様の手法により業務処理インタフェース 4 1 にプラグインされる。

【 0 1 7 4 】

アプリケーション 1 1 は、画面表示に関する処理のみではなく、業務処理もこの画面制御プログラム 3 9 を利用して実行可能である。

【 0 1 7 5 】

ユーザは、この業務処理用のコンポーネント 4 2 の利用を望む場合には、他の画面表示に関わるコンポーネントと同様に、コンポーネント 4 2 を指定し、必要な属性を指定するのみでよい。

【 0 1 7 6 】

これにより、画面プログラムの開発のみではなく、アプリケーションの開発も効率化できる。

【0177】

なお、上記各実施の形態で説明した発明は、それぞれを自由に組み合わせることができる。また、上記各実施の形態で説明した各機能、各要素は、同様の作用・機能を実現可能であれば配置を変更させてもよく、また各機能、各要素を自由に組み合わせてもよい。

【0178】

上記各実施の形態で説明した画面制御プログラムの各機能、各要素は、コンピュータに実行させることのできるプログラムであり、例えば磁気ディスク（フロッピーディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体に書き込んで適用したり、通信媒体により伝送して、計算機、計算機システムに適用可能である。

【0179】

上記各実施の形態で説明した画面制御プログラムの各機能を実現するコンピュータは、記録媒体に記録されたプログラムを読み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

【0180】

【発明の効果】

以上詳記したように本発明においては、動的に変化する表示内容を得るために予め作成されている動的表示処理を指定し、この動的表示処理への属性を指定するのみで、動的な表示内容を取得し画面に表示させることができる。

【0181】

したがって、画面プログラムの開発や変更においてプログラミングする必要がなく、コンパイルする必要もない。ゆえに、ユーザは画面プログラムを効率的かつ容易に開発・変更でき、ユーザの労力が軽減される。

【0182】

また、本発明においては、正常な入力を保持しておき、障害発生後にはこの保持しておいた内容に基づいて、入力内容が再現される。

【 0 1 8 3 】

したがって、ユーザの操作が効率化され、容易に画面プログラムの耐障害性を向上させることができる。

【図面の簡単な説明】

【図 1】

本発明の第 1 の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図 2】

レイアウト情報の具体例を示す図。

【図 3】

同実施の形態に係る画面制御プログラムがレイアウト情報を読み込んだ場合の処理手順を例示する図。

【図 4】

同実施の形態において〈field〉タグの数だけ繰り返される各フィールドの作成処理を例示するフローチャート。

【図 5】

レイアウト情報に基づく制御機能の処理によって構成される画面を例示する図。

【図 6】

同実施の形態において画面制御プログラムがレイアウト情報を読み込んだ場合の処理手順の変形例を示す図。

【図 7】

レイアウト情報において定義された入力チェック属性に従って画面制御プログラムが実行する入力チェック処理の処理手順を例示する図。

【図 8】

入力チェックの結果がエラーの場合の画面を例示する図。

【図 9】

動的表示情報の具体例を示す図。

【図 1 0】

同実施の形態に係る画面制御プログラムが動的表示情報を読み込んだ場合の処理手順を例示する図。

【図 1 1】

動的表示情報に基づく制御機能の処理によって構成される画面を例示する図。

【図 1 2】

同実施の形態において画面制御プログラムが動的表示情報を読み込んだ場合の処理手順の変形例を示す図。

【図 1 3】

同実施の形態においてフィールドにセットされた表示属性が検索だった場合に画面制御プログラムが実行する処理の手順を例示する図。

【図 1 4】

同実施の形態においてフィールドにセットされた表示属性が検索だった場合の画面制御プログラムの処理を例示するフローチャート。

【図 1 5】

同実施の形態において検索処理用のコンポーネントによって動的な表示内容が取得され、検索された一覧がユーザの操作に基づいて表示された画面を例示する図。

【図 1 6】

同実施の形態においてフィールドにセットされた表示属性がポップアップだった場合に画面制御プログラムが実行する処理の手順を例示する図。

【図 1 7】

同実施の形態において、ユーザがフィールドに対して操作を行った結果、起動されるポップアップ処理の手順を例示する図。

【図 1 8】

ポップアップウィンドウが表示されている画面を示す図。

【図 1 9】

ポップアップウィンドウで選択されたデータがフィールドにセットされた画面を示す図。

【図 2 0】

本実施の形態において画面表示トランザクション機能を利用して画面表示を復帰させる処理の手順を例示する図。

【図 2 1】

本発明の第 2 の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図 2 2】

同実施の形態に係る画面制御プログラムがレイアウト情報を読み込んだ場合の処理手順を例示する図。

【図 2 3】

本発明の第 3 の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図 2 4】

本発明の第 4 の実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図。

【図 2 5】

本発明の第 5 の実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図。

【図 2 6】

本発明の第 6 の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【符号の説明】

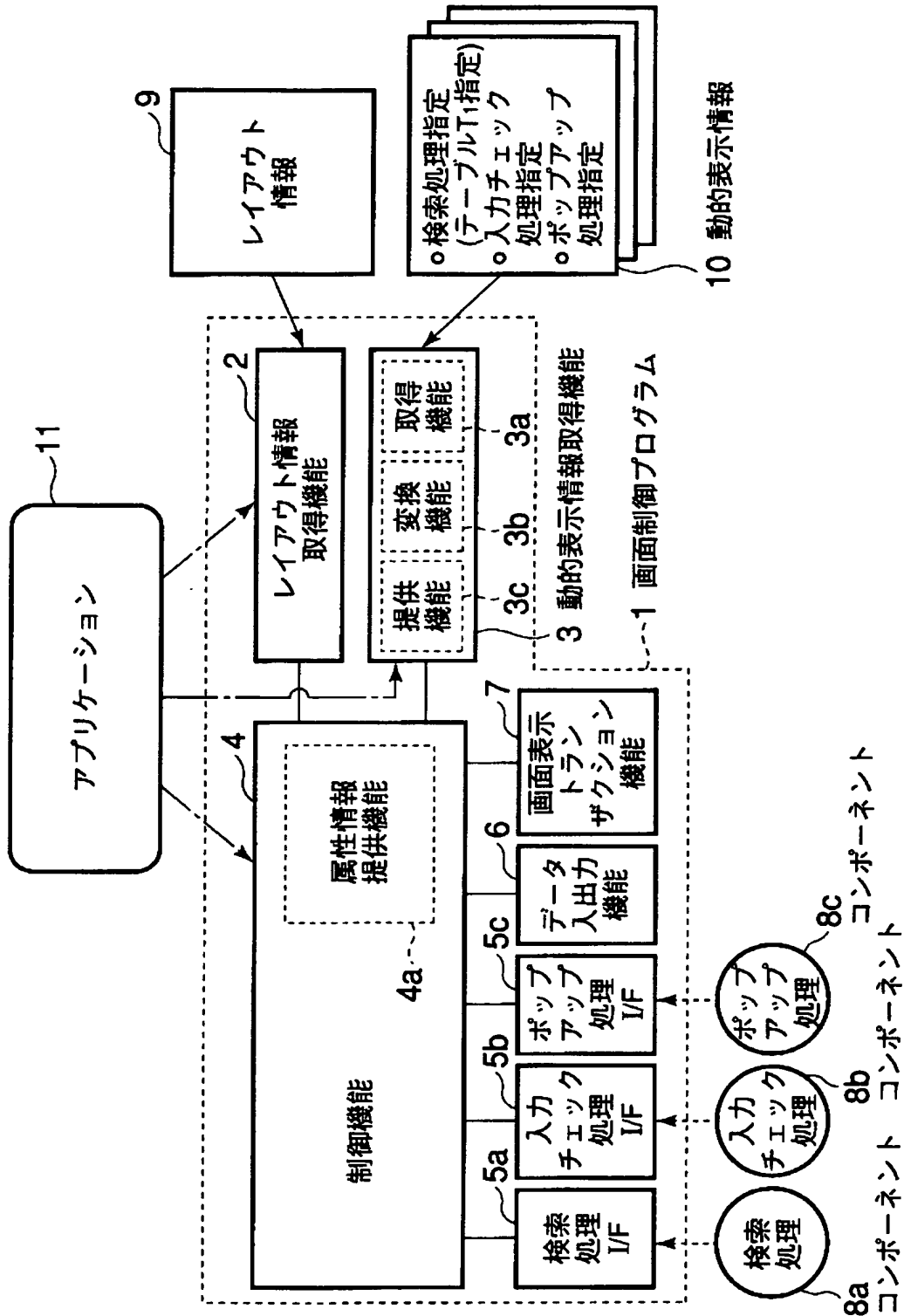
- 1、1 7、2 0、2 6、3 4、3 9…画面制御プログラム
- 2、1 8、2 4…レイアウト情報取得機能
- 3、2 5…動的表示情報取得機能
- 3 a…取得機能
- 3 b…変換機能
- 3 c…提供機能
- 4、4 0…制御機能
- 4 a…属性情報提供機能

- 5 a、2 7…検索処理インタフェース
- 5 b、3 1…入力チェック処理インタフェース
- 5 c…ポップアップ処理インタフェース
- 6…データ入出力機能
- 7、3 5…画面表示トランザクション機能
- 8 a～8 c…コンポーネント
- 9、1 9…レイアウト情報
- 1 0…動的表示情報
- 1 1…アプリケーション
- 2 1…計算機
- 2 2、2 9、3 7…ネットワーク
- 2 3…WWWサーバ
- 2 8…検索処理スタブ
- 3 0…検索処理リモートオブジェクト
- 3 2…入力チェック処理スタブ
- 3 3…入力チェック処理リモートオブジェクト
- 3 6…画面表示トランザクション情報送受信部
- 3 8…画面表示トランザクション管理サーバ
- 4 1…業務処理インタフェース
- 4 2…業務処理

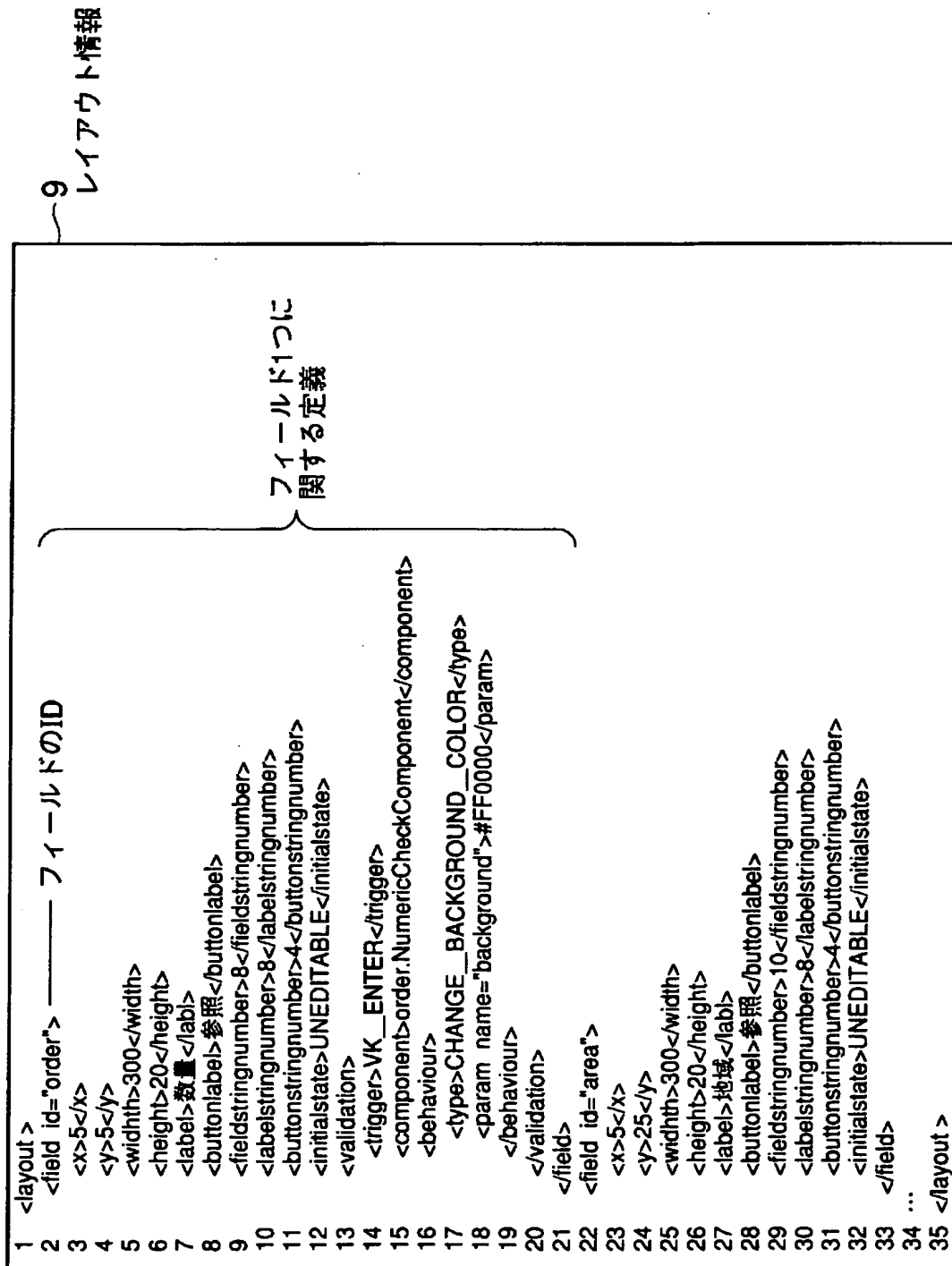
【書類名】

図面

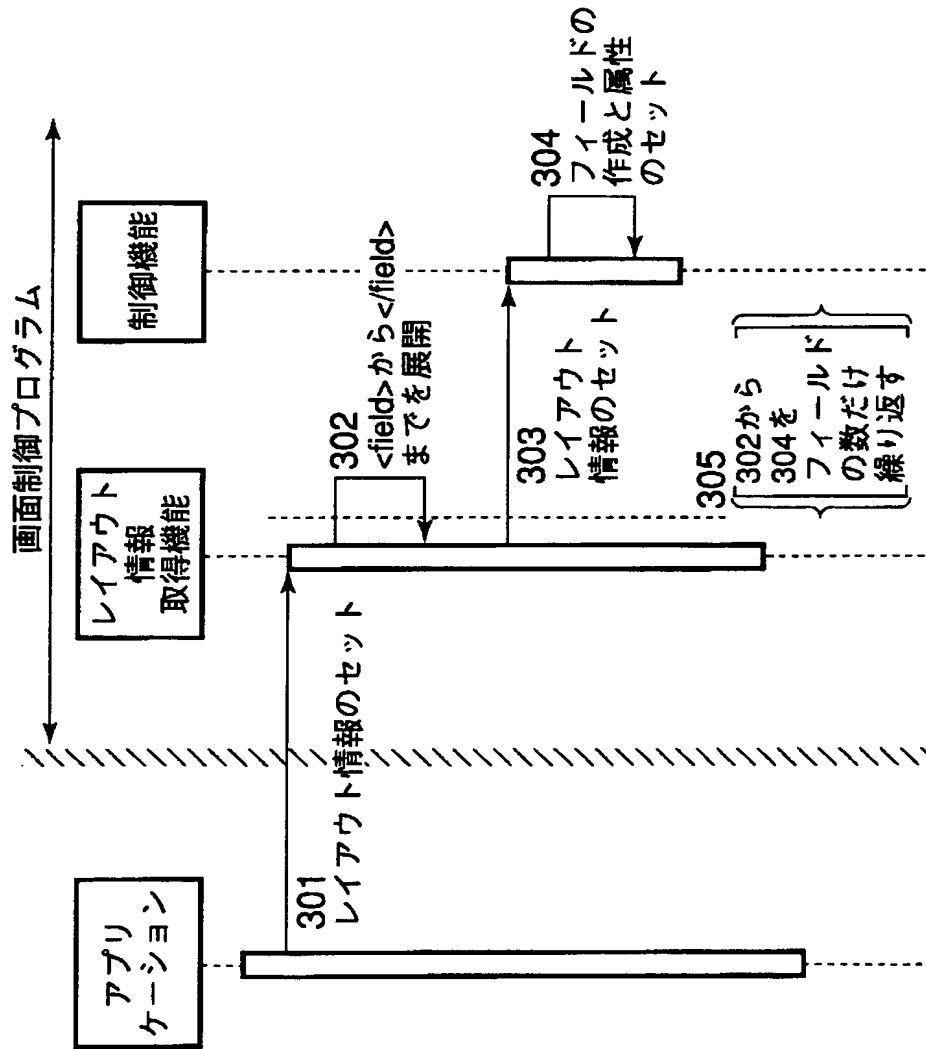
【図 1】



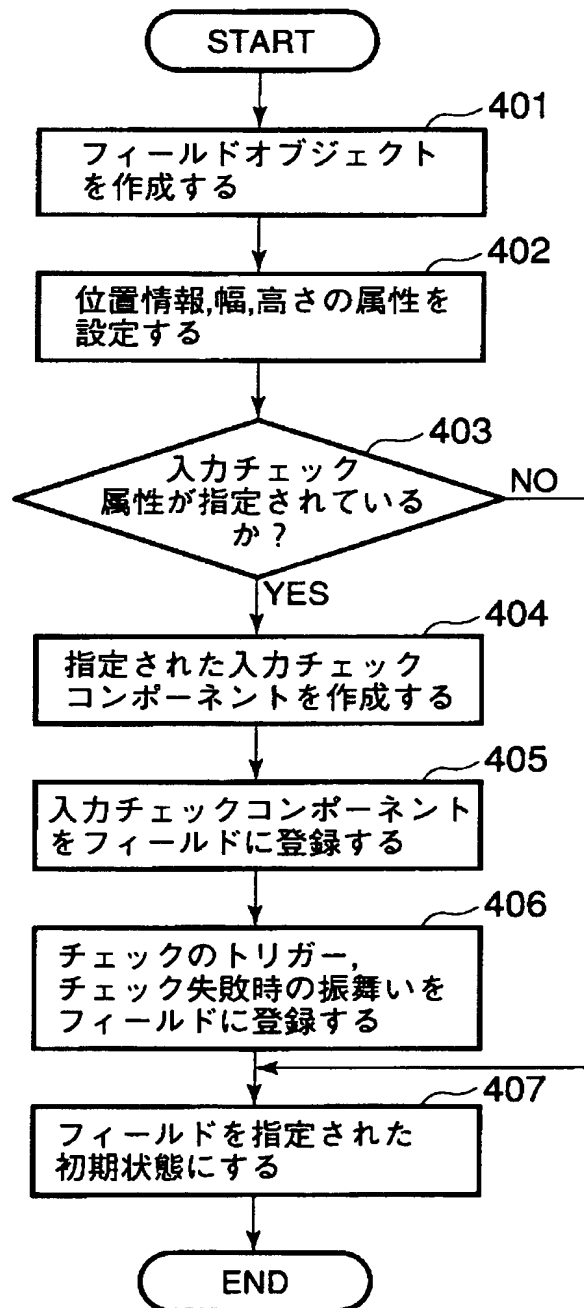
【図 2】



【図 3】



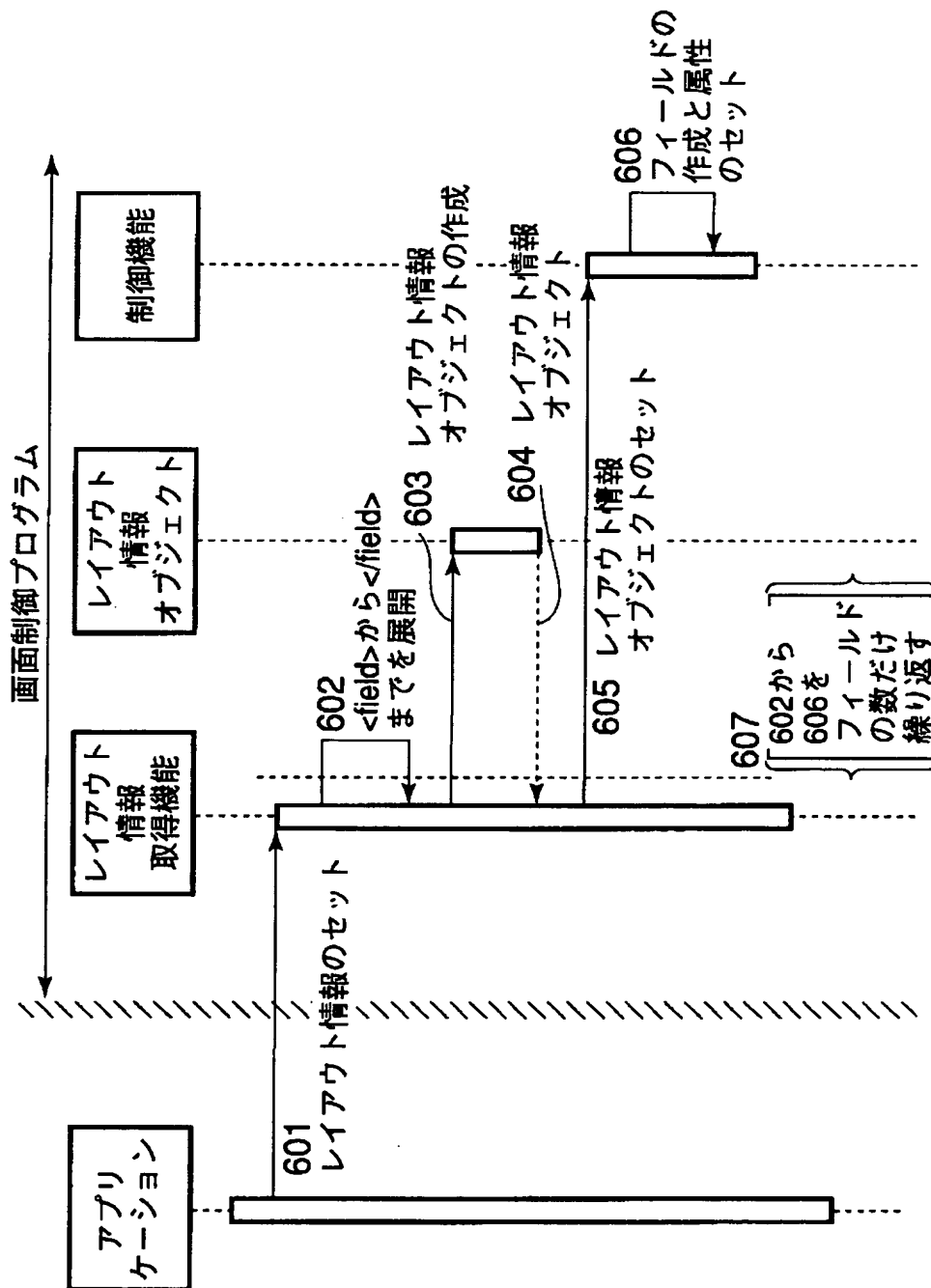
【図 4】



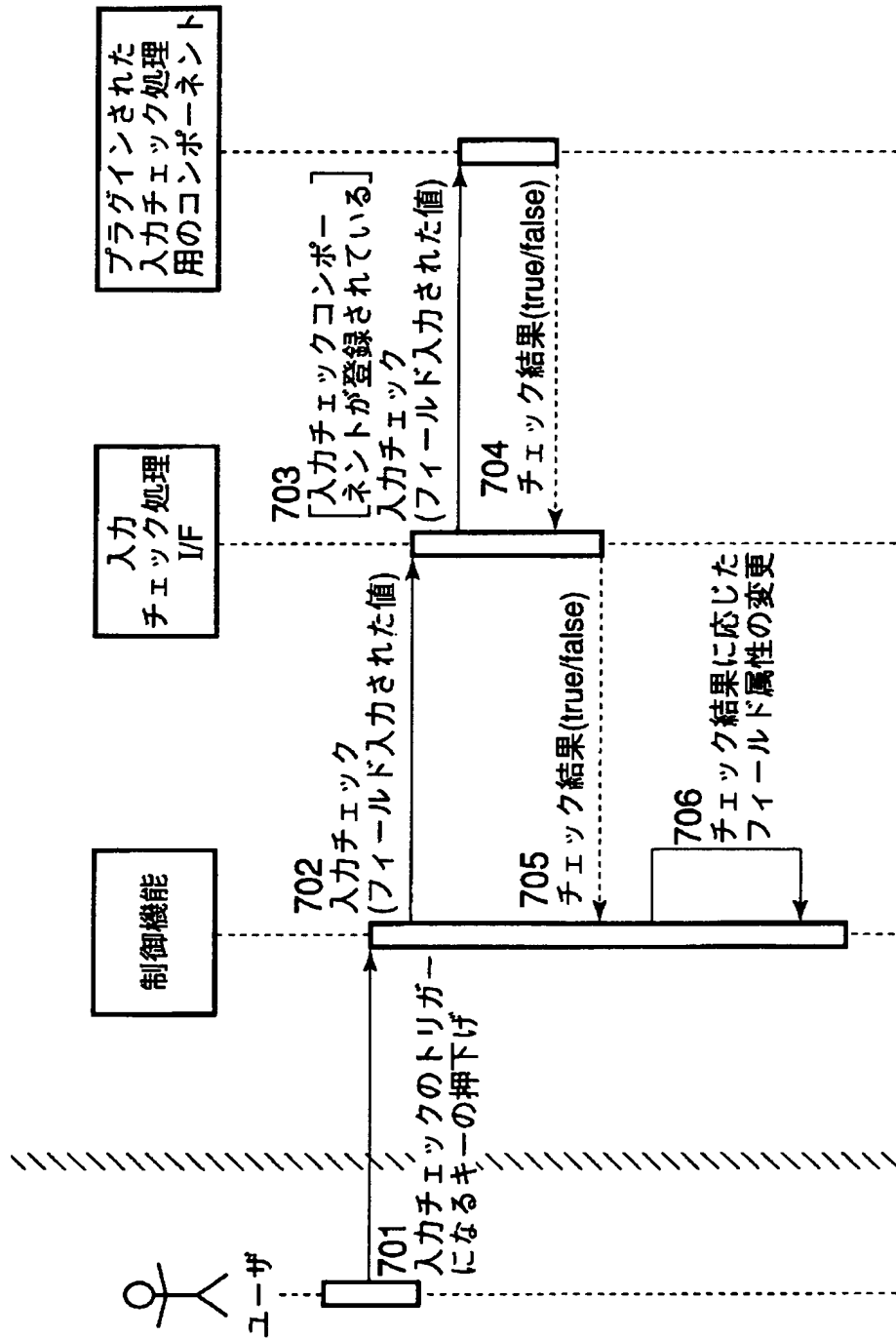
【図 5】

The diagram illustrates a software screen (12画面) with a standard window interface. The title bar at the top contains three small, empty square icons. The main content area features three vertically stacked dashed rectangular input fields. To the left of these fields are labels: '数量' (Quantity) for the top field, '地域' (Region) for the middle field, and '商品' (Goods) for the bottom field. Each label is preceded by a reference numeral: '12a' for '数量', '12b' for '地域', and '12c' for '商品'. To the right of each dashed field is another reference numeral: '12d' for the top field, '12e' for the middle field, and '12f' for the bottom field. At the bottom right of the screen, there are three rectangular buttons arranged horizontally, labeled 'OK', 'キャンセル' (Cancel), and '終了' (End). A bracket at the bottom left of the screen frame points to the label '12 画面'.

【図 6】



【図 7】



【図 8】

数量 12d

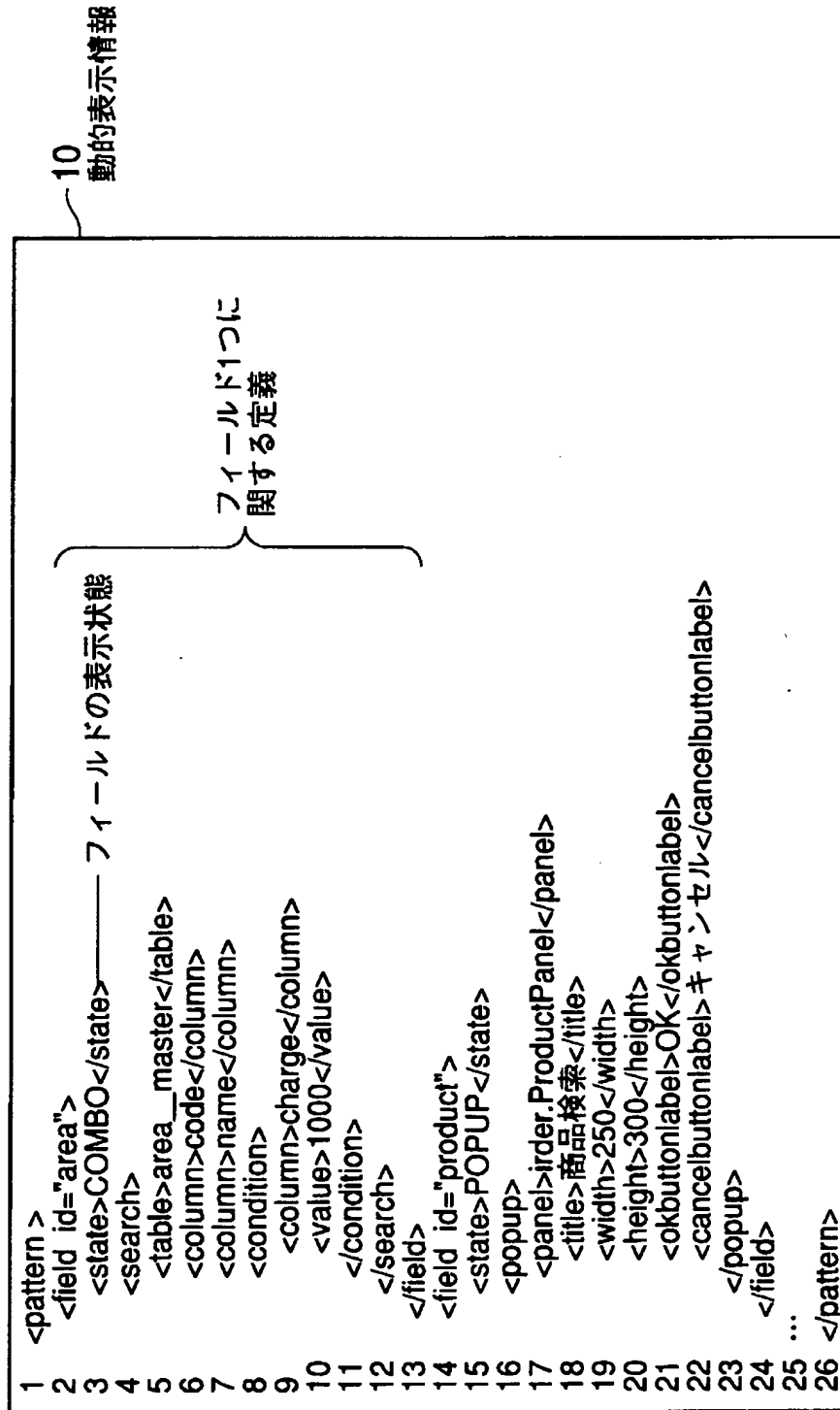
地域 ▼

商品 参照

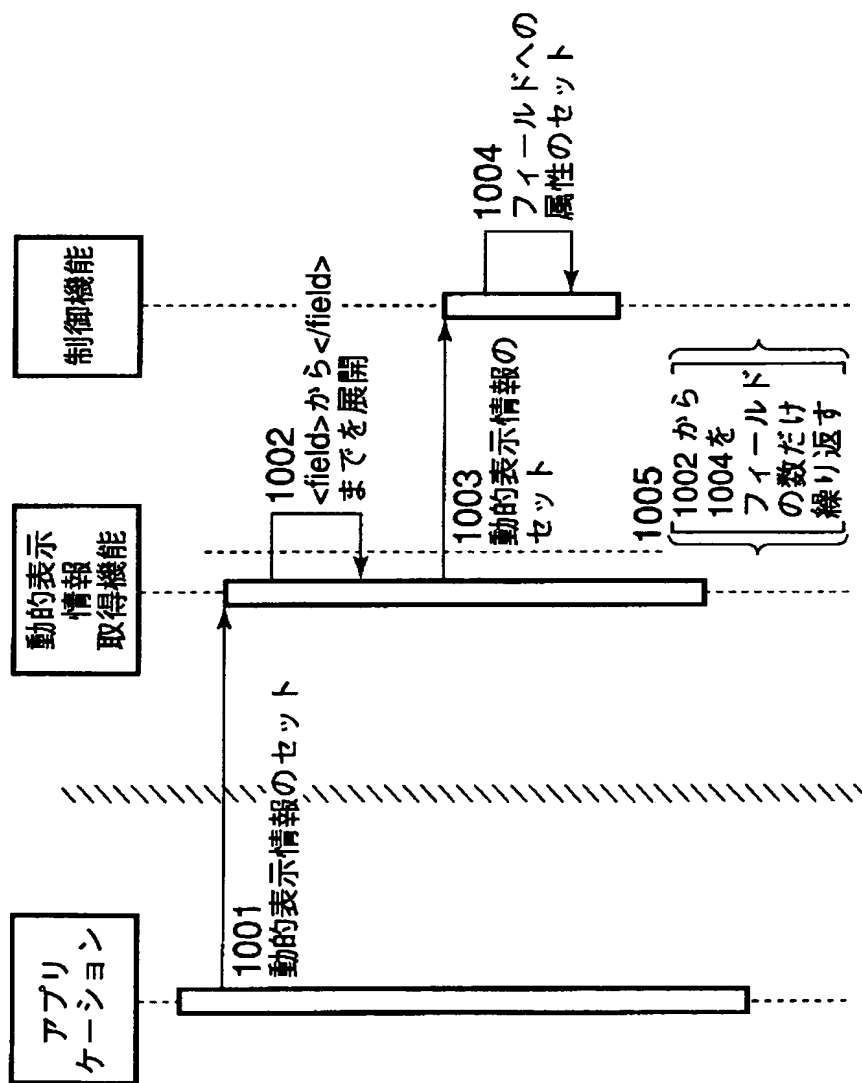
OK キャンセル 終了

12 画面

【図 9】



【図 1 0】



【図 1 1】

The figure shows a software window with a title bar at the top containing three small icons. The main area of the window contains three input fields arranged vertically. The first field is labeled '数量' (Quantity) and is empty. The second field is labeled '地域' (Region) and is empty, with a downward-pointing triangle icon to its right, labeled '14a'. The third field is labeled '商品' (Product) and is empty, with a button labeled '参照' (Reference) to its right, labeled '14b'. At the bottom right of the window, there are three buttons: 'OK', 'キャンセル' (Cancel), and '終了' (End). A label '12 画面' (Screen 12) points to the bottom left corner of the window.

数量

地域

商品

参照

OK

キャンセル

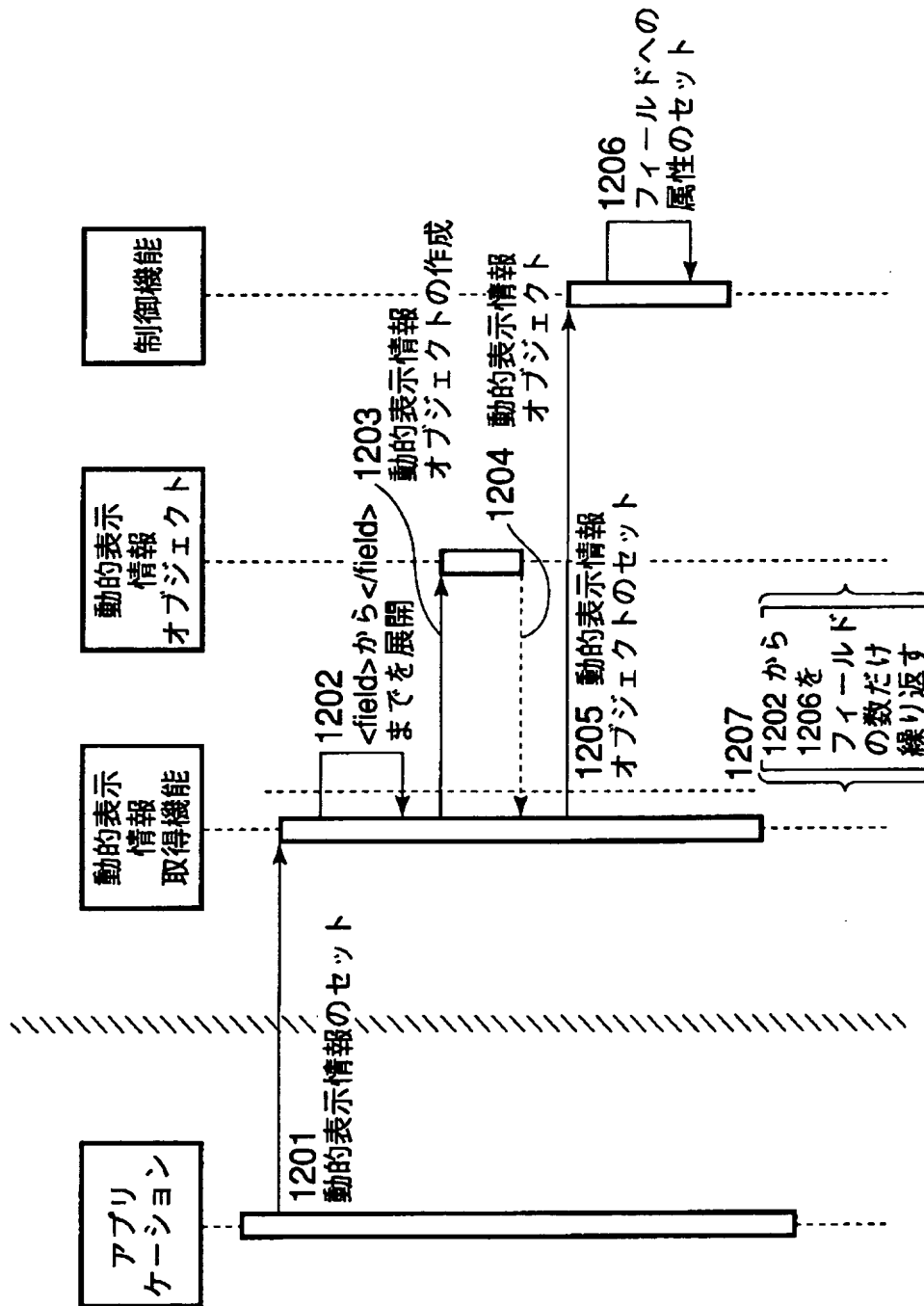
終了

12 画面

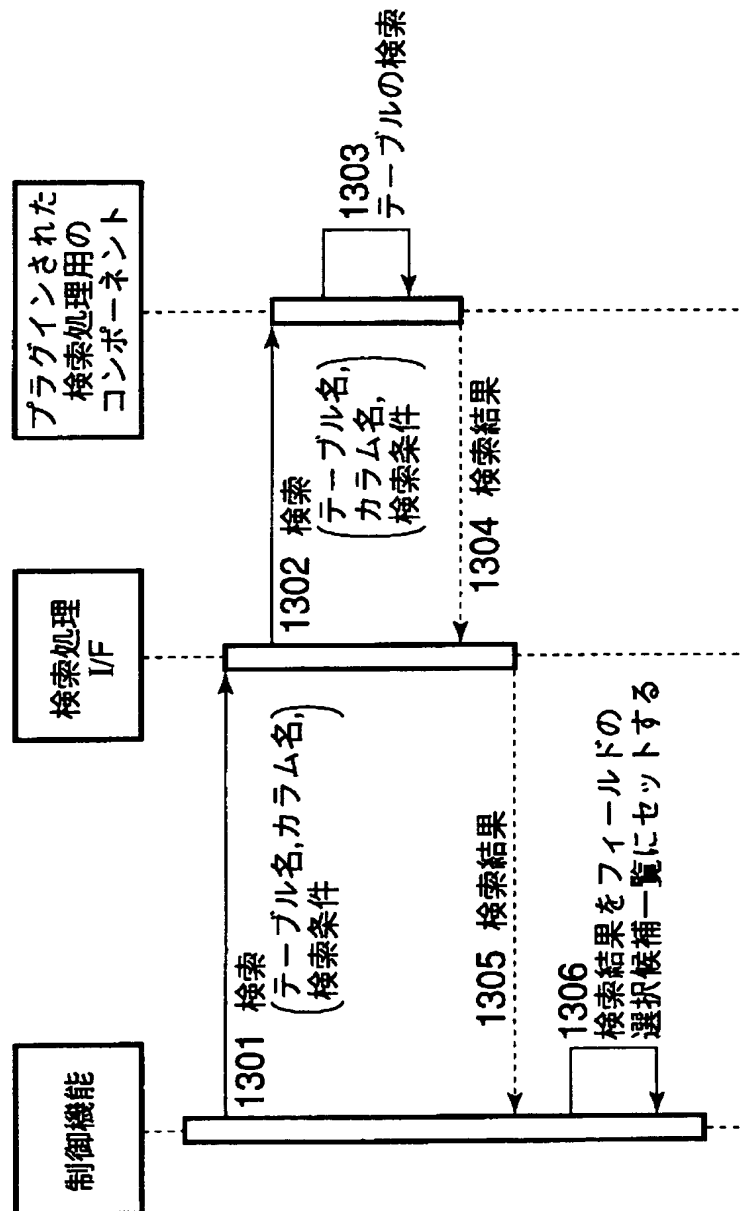
14a

14b

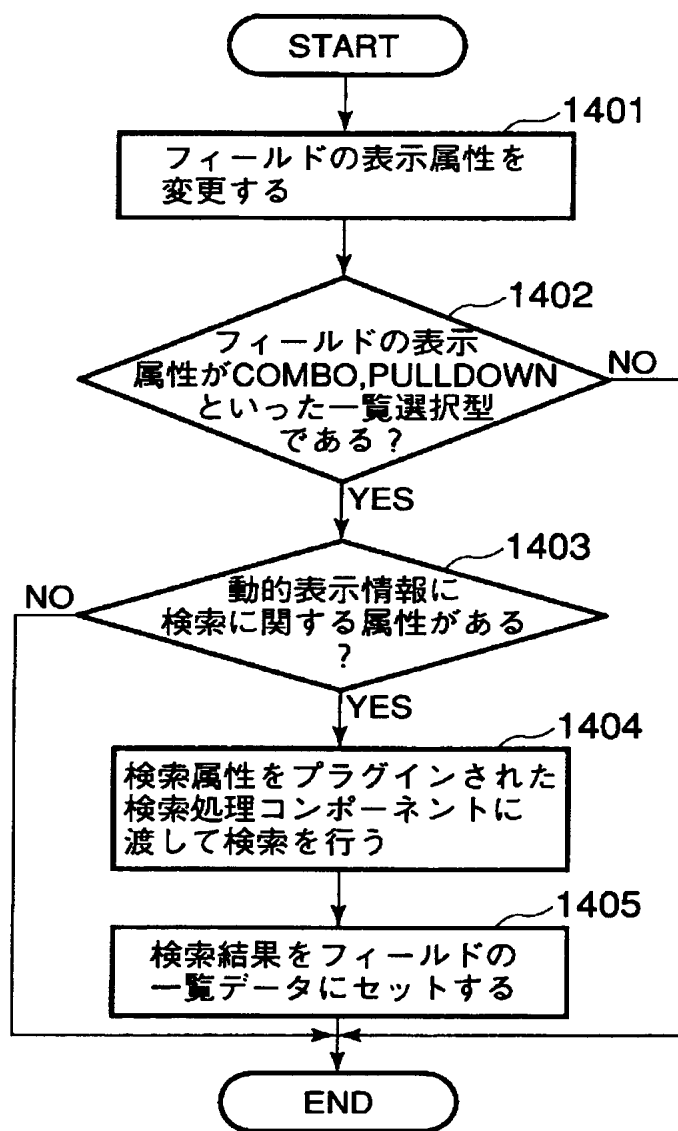
【図 1 2】



【図 1 3】



【図 1 4】



【図 1 5】

数量

地域

商品

001 北海道
002 青森
003 岩手
004 秋田

参照

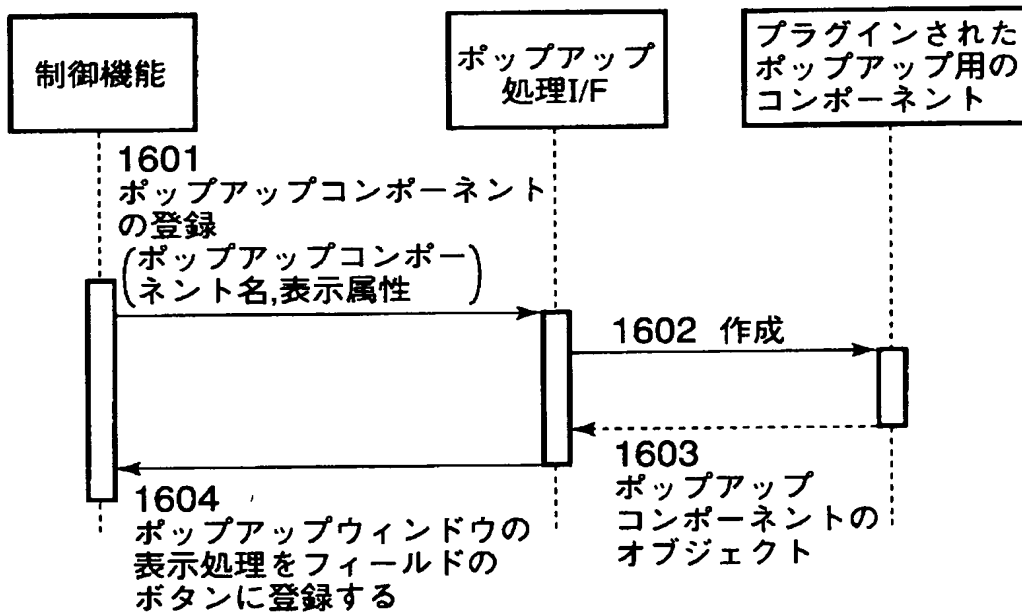
OK キャンセル 終了

12 画面

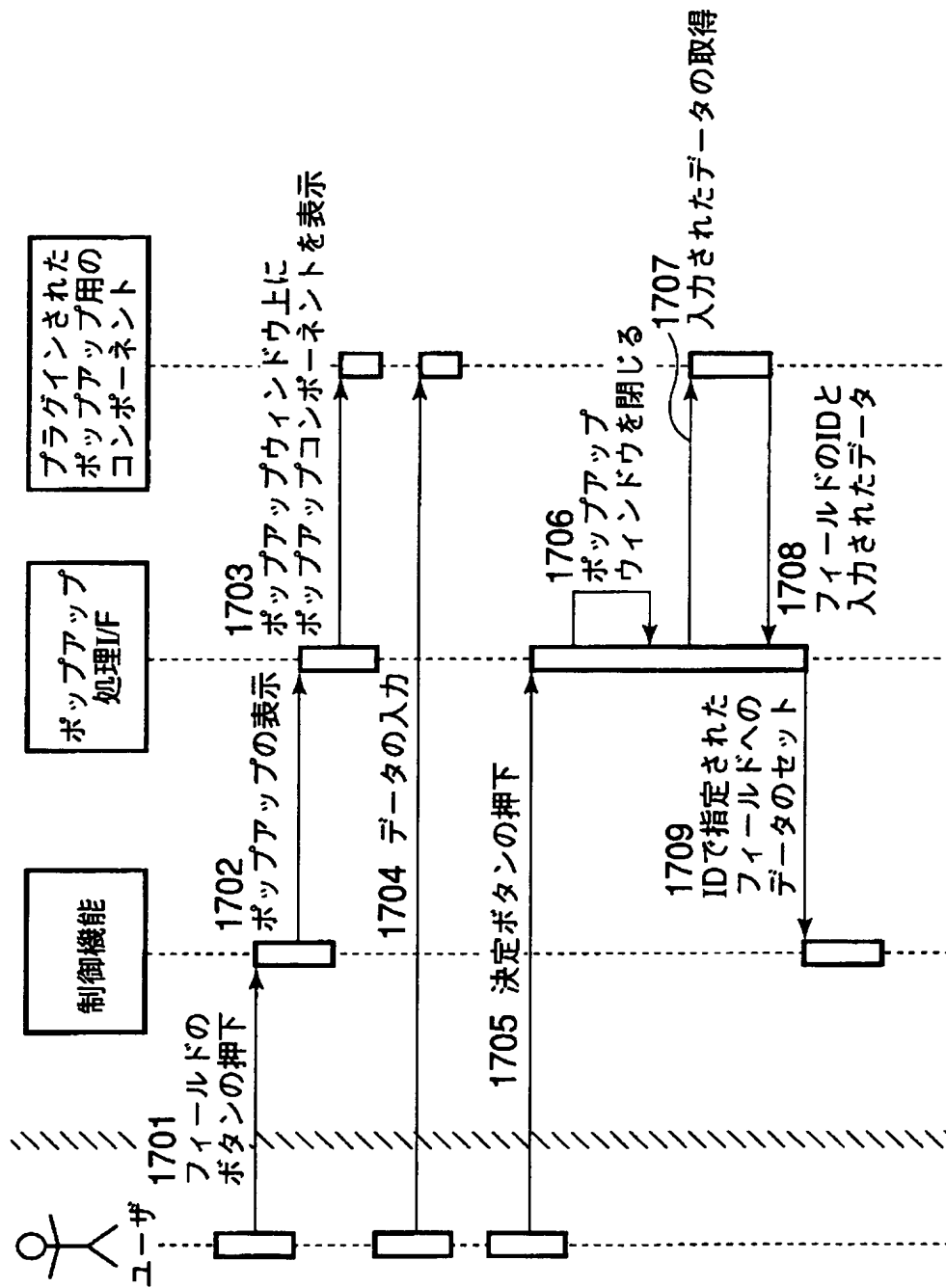
12e

15a

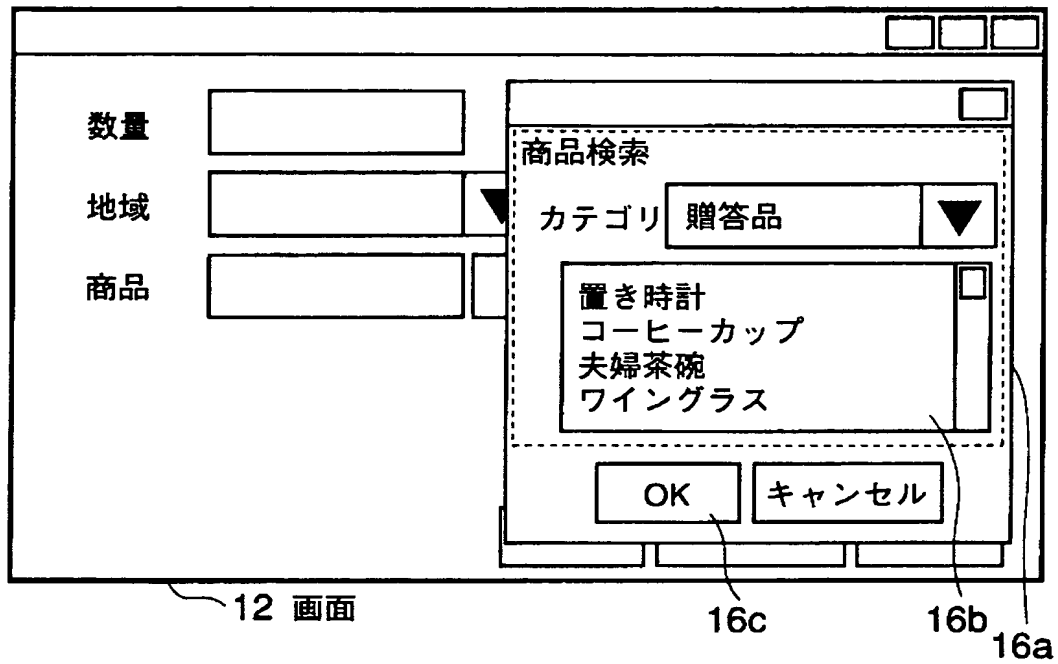
【図 1 6】



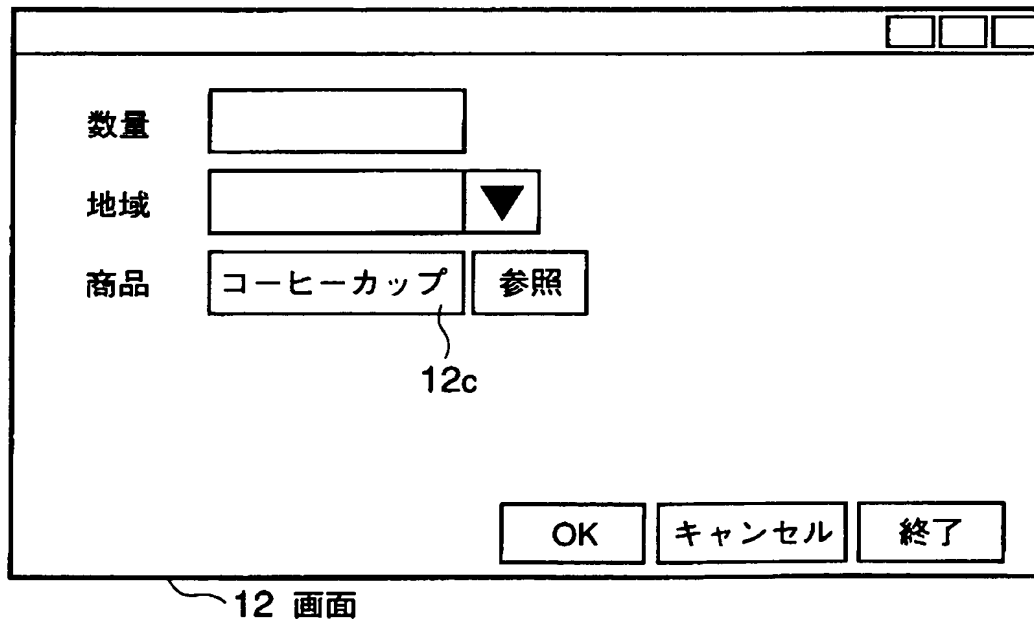
【図 1 7】



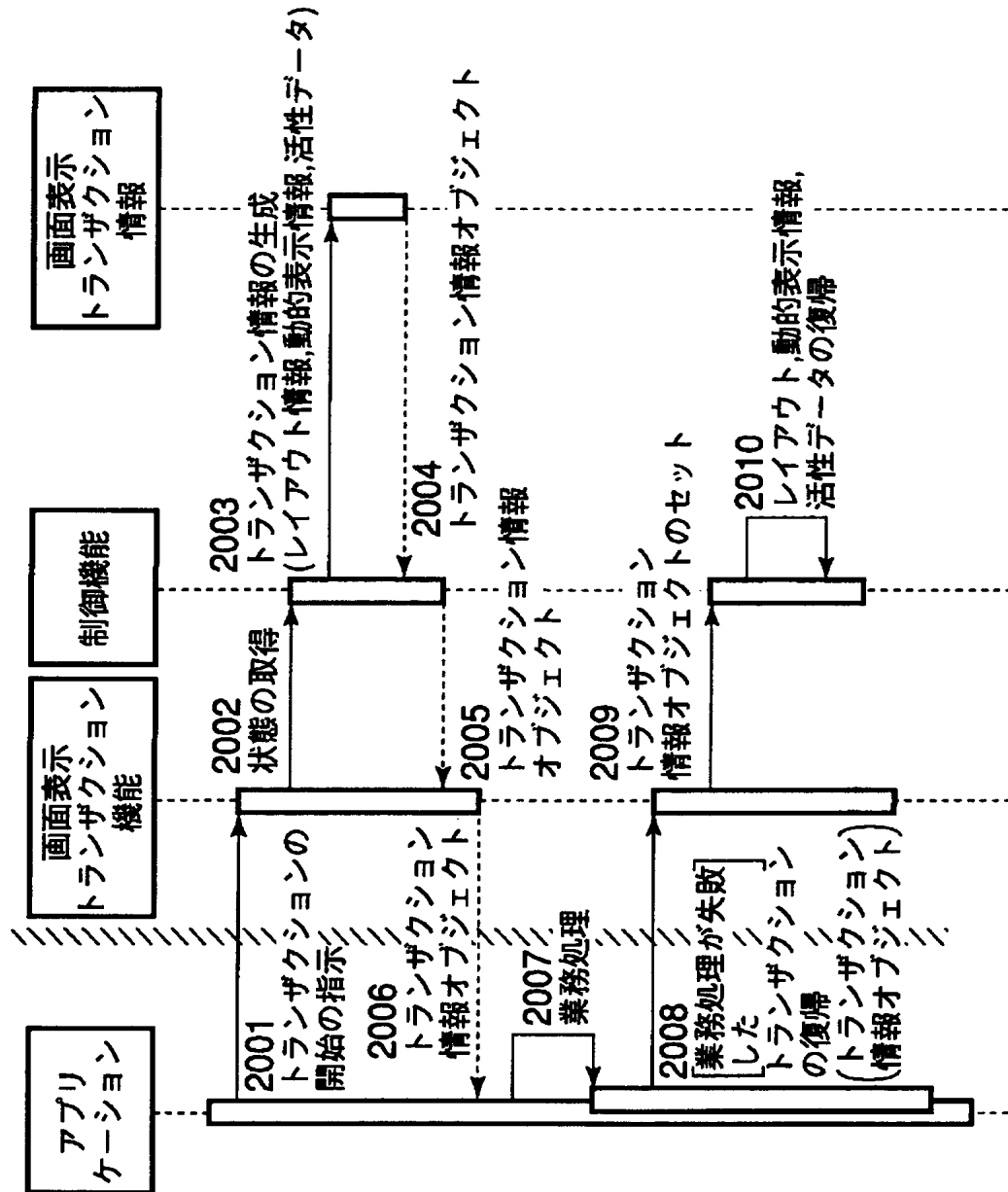
【図 1 8】



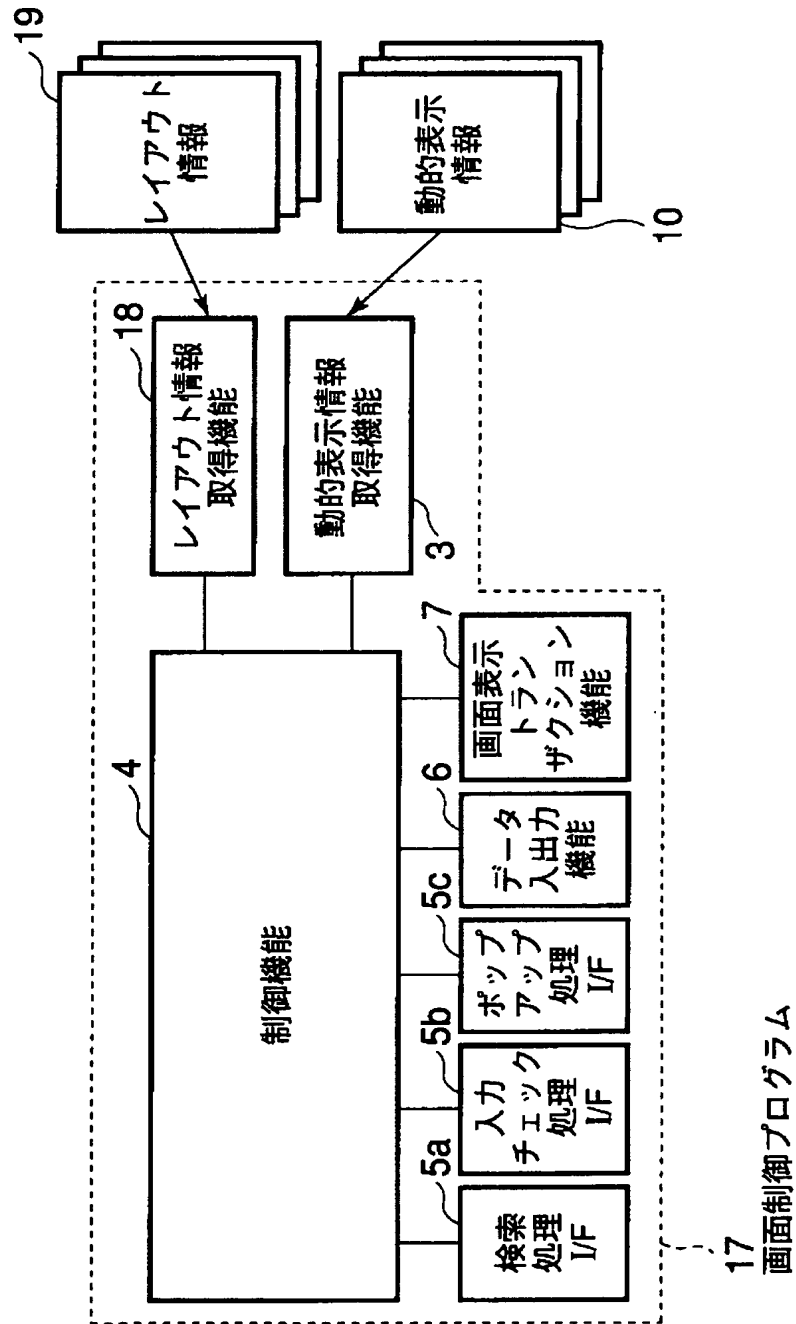
【図 1 9】



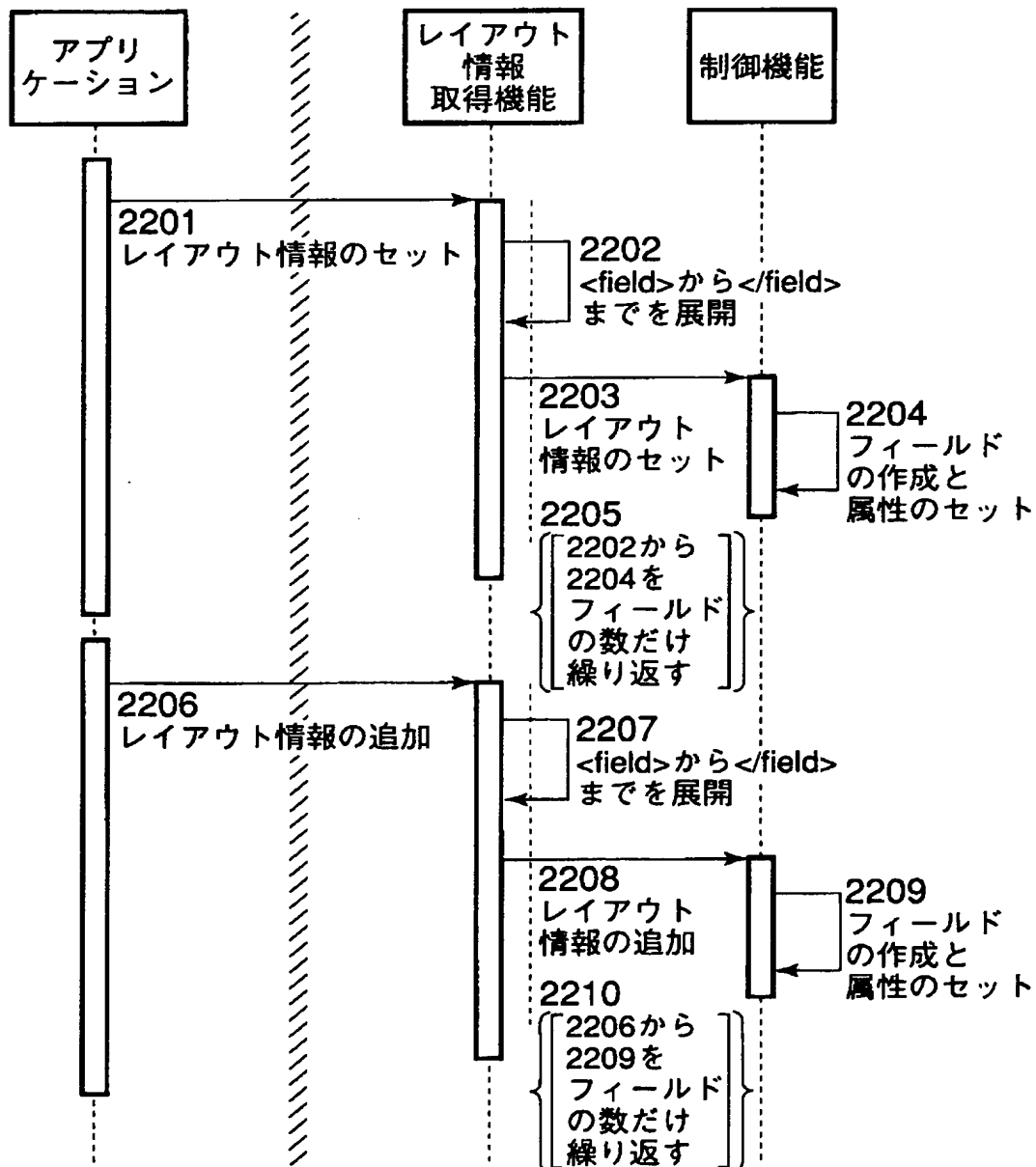
【図 20】



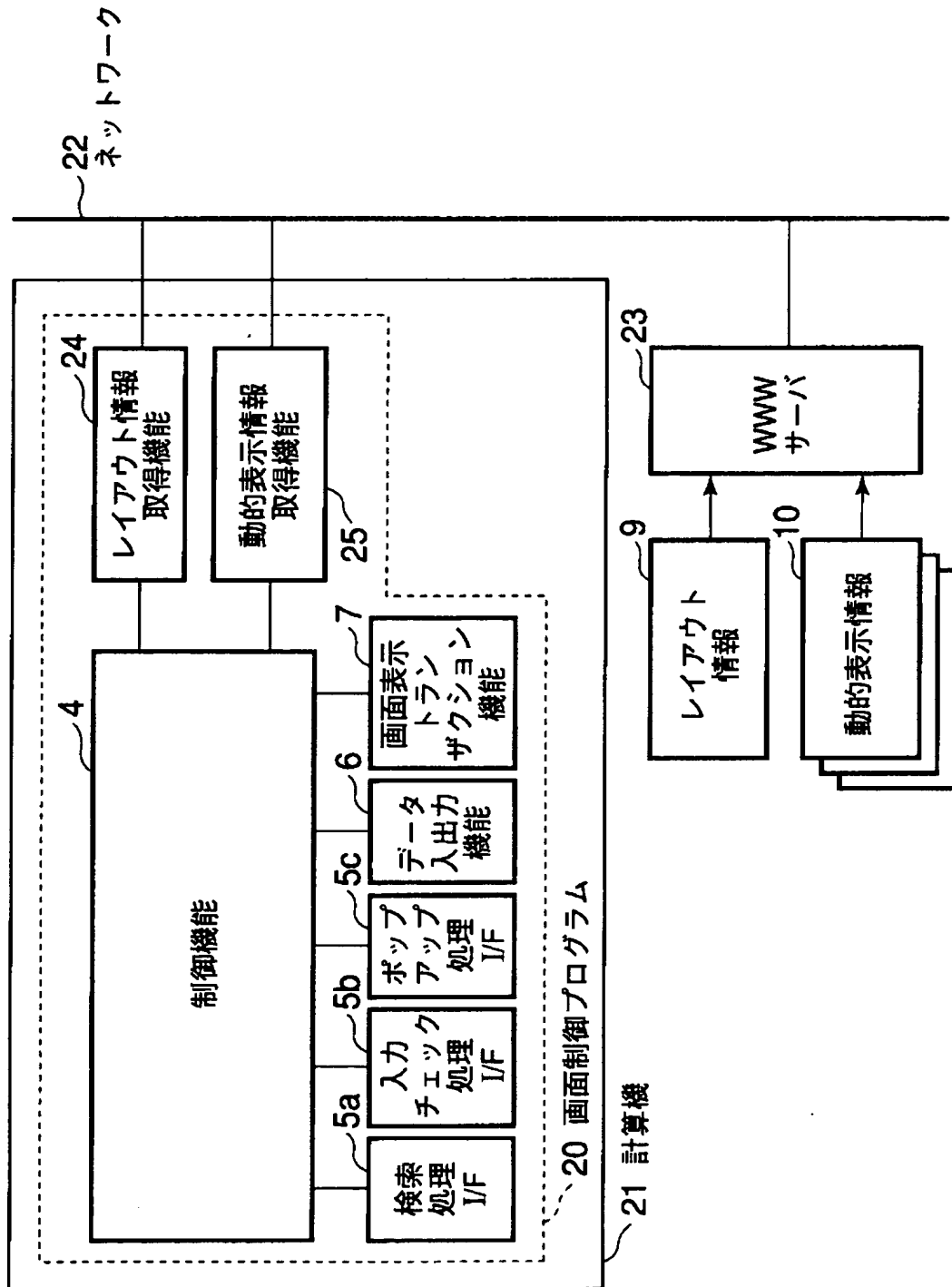
【図 2 1】



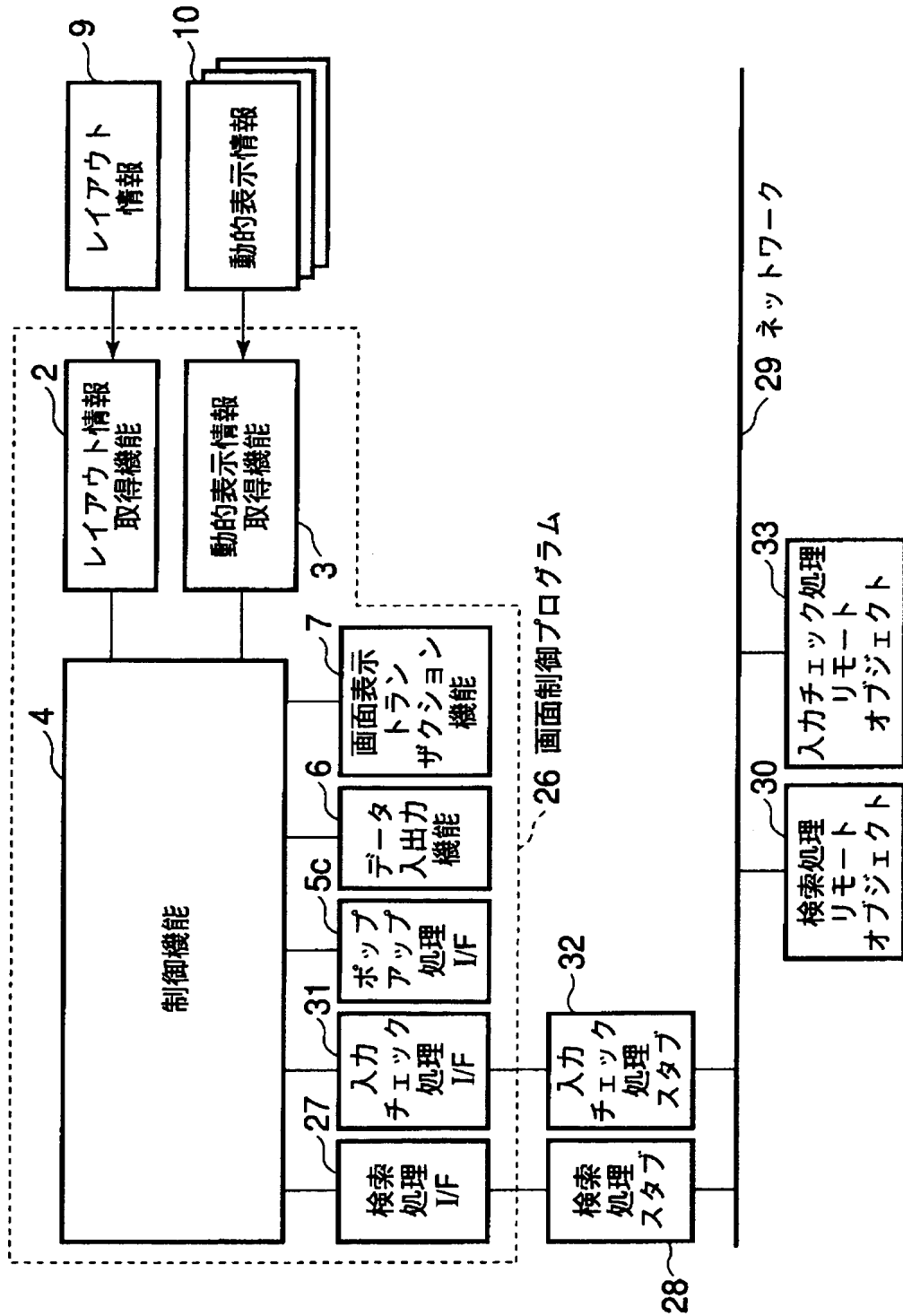
【図 2 2】



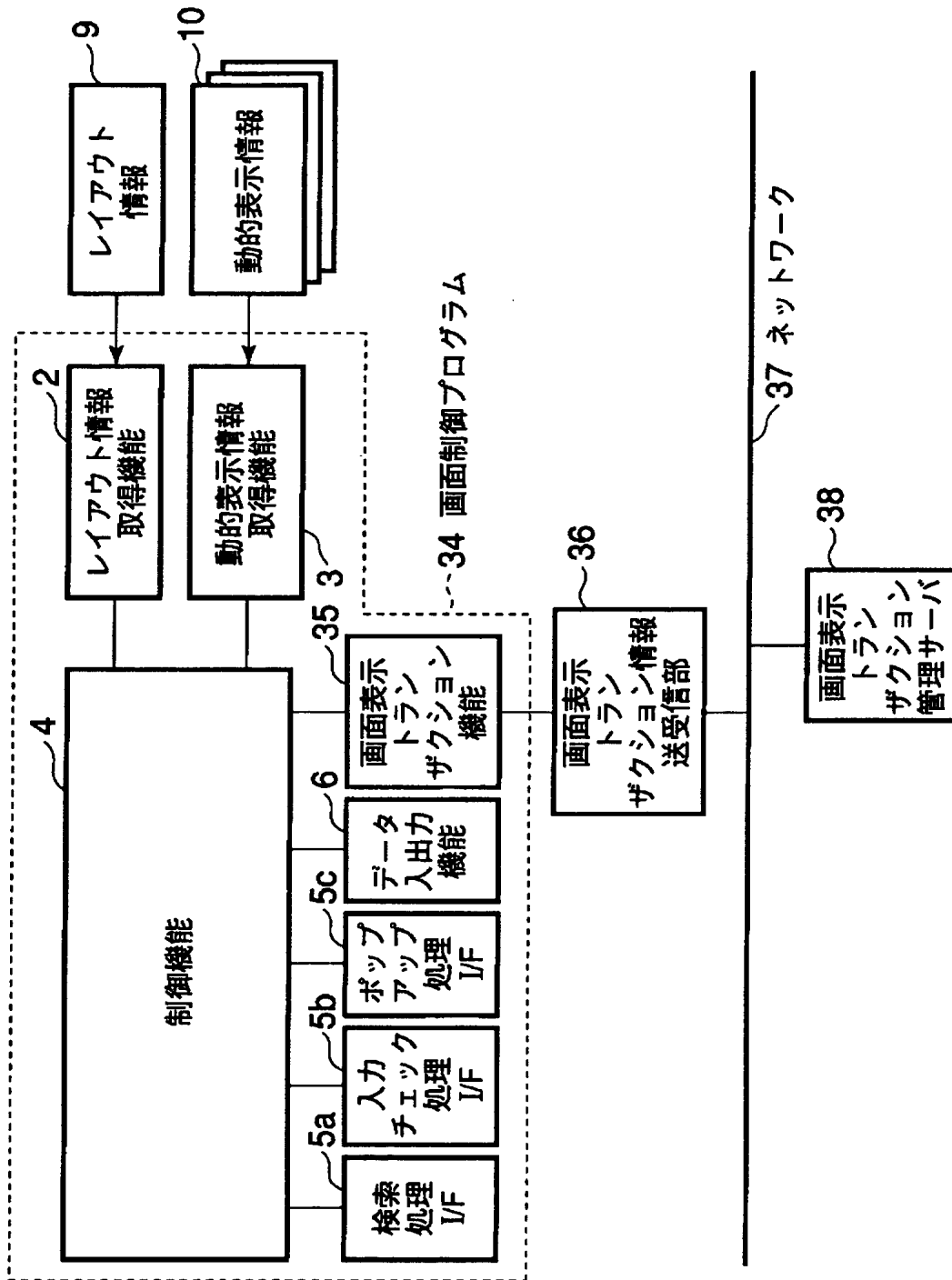
【図 2 3】



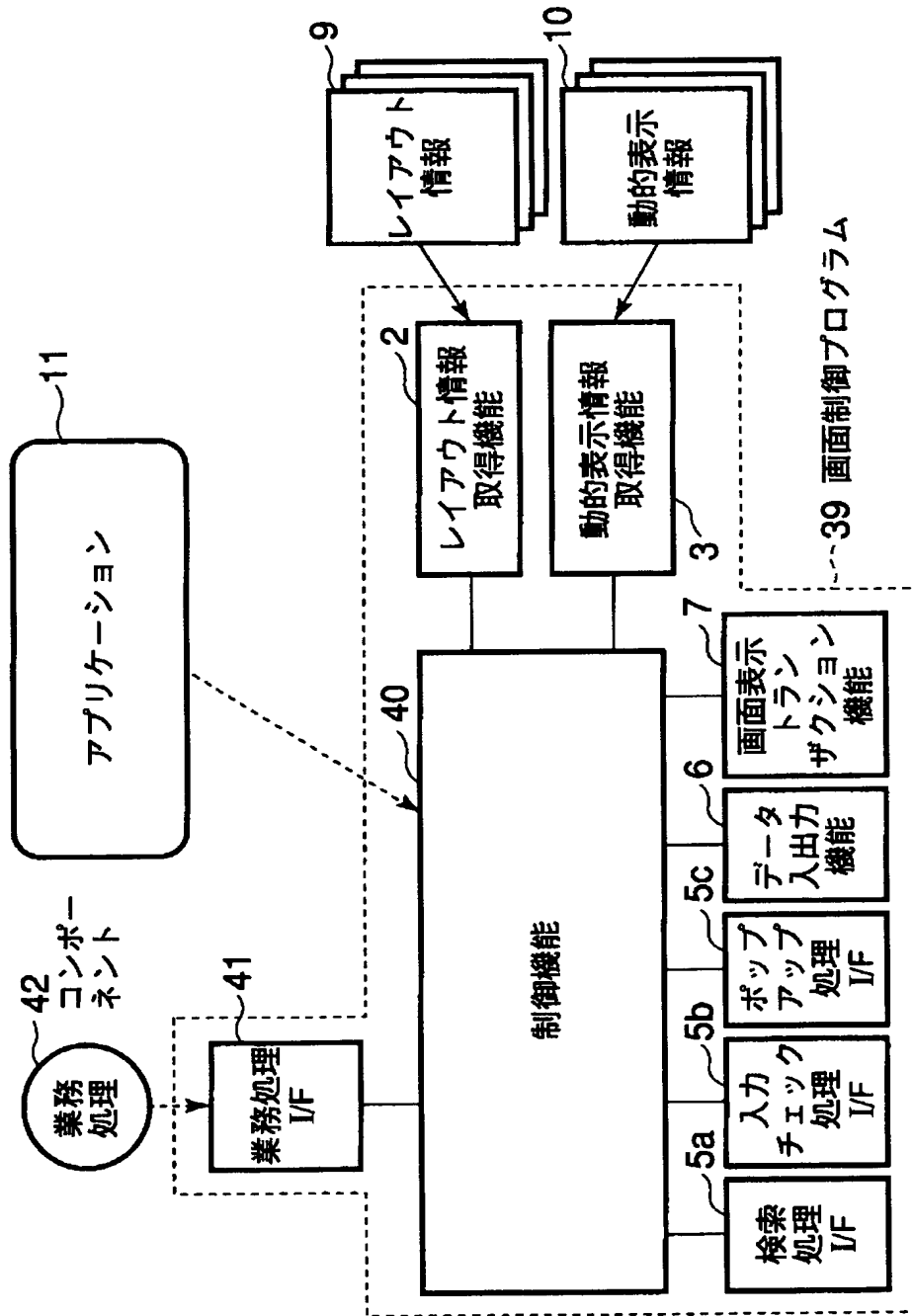
【図 24】



【図 2 5】



【図 26】



【書類名】 要約書

【要約】

【課題】画面プログラムの開発・変更を効率化する。

【解決手段】コンピュータに、画面に表示される場合に動的に変化する表示内容を得るために予め作成されている少なくとも一つの動的表示処理 8 a ～ 8 c のうちのいずれかを指定する動的表示処理識別情報とこの動的表示処理識別情報で指定されている動的表示処理に提供される表示属性情報とを取得させる動的表示情報取得機能 3 と、動的表示情報取得機能 3 によって取得された動的表示処理識別情報で指定されている動的表示処理に対して該当する表示属性情報を提供させる属性情報提供機能 4 a とを実現させる画面制御プログラム 1 を記録したコンピュータ読み取り可能な記録媒体を提供する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 3 0 7 8]

1. 変更年月日 1 9 9 0 年 8 月 2 2 日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町 7 2 番地

氏 名 株式会社東芝